News From The Summer Consumer Electronics Show

COMPUTE!

$2.95
August
1985
Issue 63
Vol. 7, No. 8
$3.75 Canada
02193
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing

Animator
The Easy Way To Create Your Own Colorful Graphics On IBM, Commodore, Apple, Atari, And TI

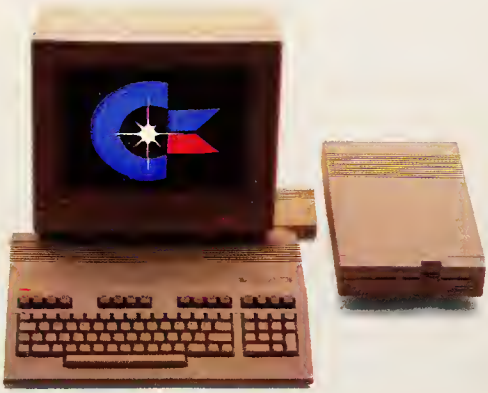
Advanced Commodore 1541 Disk Commands
Add Power To Your Programs

Apple IIc Mousor
Efficient Editing With Fast Cursor Control

Atari Color Mixing
How To Create Dazzling Color Displays

IBM Filecopy
A Better Way To Make Backups

If you own an Apple IIc,
you'd have to add three more Apple IIc's,
an Extra Keypad,
30 Block Graphic Sets,
Color Sprites,
two more voices,
four instruments,
a Cartridge Port, a Joystick Port,
and a Commodore 64...



to match the versatility, expandability and higher intelligence of the new Commodore 128

(and it costs less too).

The new Commodore 128™ personal computer is breakthrough technology at a breakthrough price. It outshines the Apple® IIc in performance capability, performance quality and price. It is expandable to 512K RAM. The IIc doesn't expand. Commodore 128 has a numeric keypad built into its keyboard that makes crunching numbers a lot easier. And graphic and sound capabilities that far exceed those of the Apple IIc. But the most important news is that Commodore 128 jumps you into a new world of business, productivity, education and word processing programs while still running over 3,000 programs designed for the Commodore 64.™ That's what we call a higher intelligence.

COMMODORE 128 PERSONAL COMPUTER

A Higher Intelligence

*Apple is a registered trademark of Apple Computer, Inc.

AMAZING DAISY

NOW! FULL SIZE, FULL FEATURE, LETTER QUALITY AT ONLY $353

If you have been searching for a letter quality printer you have probably found the flood of claims and counterclaims to be a real roadblock in your search. Not long ago we were in the same position. We tried to determine which daisy wheel printer had all the features our customers wanted, yet would not set them back a month's salary. Recently several manufacturers have introduced machines that had features we were seaching for. After a thorough assessment, we eliminated one model after the other for lack of one feature or another until we only had one left.

THE RESULTS ARE IN

We found the printer which has all the features anyone could want. The winner is the Aprotek Daisy 1120, a real heavy-duty workhorse printing at 20 characters per second. The manufacturer is Olympic Co. Ltd., a highly respected Japanese firm.

FEATURES GALORE

This printer has it all. To start with, it has a front panel Pitch Selector button with indicators which allows 10, 12, 15 characters per inch (CPI) or Proportional Spacing. There is a Select (Online) button (with indicator) and a Line Feed button. You can also set Top-of-Form or Form Feed with the touch of the TOF button. Other front panel indicators include Power and Alarm.

To load a sheet of paper, simply place it in the feed slot and pull the paper bail lever. PRESTO! The paper feeds automatically to a 1 inch top margin and the carriage aligns to the selected left margin. In this manner, each page can have identical margins automatically. You can continue to compute while the Daisy 1120 is printing. The built in 2K buffer frees up your computer while printing a page or two allowing you to go to your next job.

To really put your printer to work, the Cut Sheet Feeder option is great for automatic printing of those long jobs. Also available is the adjustable Tractor Feed option. Compare our option prices! Best of all the Daisy 1120 is quiet: only 57 dB-A (compare with an average of 62-65 dB-A for others).

COMPLETE COMPATIBILITY

The Daisy 1120 uses industry standard Diablo® compatible printwheels. Scores of typeface styles are available at most computer or stationary stores. You can pop in a 10, 12, 15 pitch or proportional printwheel and use paper as wide as 14". At 15 CPI you can print 165 columns—great for spreadsheets.

The Daisy 1120 uses the Diablo Hytype II® standard ribbon cartridges. Again universally available.

Not only is the hardware completely compatible, the control codes recognized by the Daisy 1120 are Diablo 630® compatible (industry standard). You can take advantage of all the great features of word processing packages like Wordstar®, pfs: Write®, Microsoft Word® and most others which allow you to automatically use superscripts, subscripts, automatic underlining, boldface (shadow printing) and doublestrike.

The printer has a set of rear switches which allow the use of standard ASCII as well as foreign character printwheels. Page length can be set to 8, 11, 12, or 15". The Daisy 1120 can also be switched to add automatic line feed if required.

THE BEST PART

When shopping for a daisy wheel printer with all these features (if you could find one), you could expect to pay $600 or $700 dollars. The options would add much more. *Not now!* We have done our homework. We can now offer this printer for only $353. Order yours today!

NO RISK OFFER

Try the Daisy 1120 for 2 weeks. If you are not satisfied for ANY reason we will refund the full price—promptly. A full 1-year parts and labor warranty is included.

THE BOTTOM LINE

Aprotek Daisy 1120 (Order#1120) $353 w/standard Centronics parallel interface and 2K buffer.

Options

Auto Cut Sheet Feeder (#1110) $188

Tractor Feed (#1112) $77

Accessories

8' Cable for IBM PC® and compatibles (#1103) $26

Interface with cable: •TI-99/4A (#106) $66
•Apple II or IIe (#1104) $76
•All Commodore (except Pet) (#1105) $44
•All Atari (#1107) $66

Shipping is $11—UPS continental USA. If you are in a hurry, UPS Blue or Air Parcel Post (second day air) is $25. Canada, Alaska, Mexico and Hawaii are $30 (air). Other foreign is $60 (air). California residents add 6% tax. Prices are cash prices—VISA and M/C add 3% to total. We ship promptly on money orders, cashier's checks, and charge cards. Allow 14-day clearing for checks. No C.O.D.'s. Payment in US dollars only.

TO ORDER ONLY CALL TOLL FREE

(800) 962-5800 USA

(800) 962-3800 CALIF. (8-8 PST)

Or send payment to address below:

Technical Information & Customer Service: (805) 987-2454 (8-5 PST)

Dealer Inquiries Invited

APROTEK

1071-A Avenida Acaso, Camarillo, CA 93010

COMPUTE!

AUGUST 1985
VOLUME 7
NUMBER 8
ISSUE 63

GUIDE TO ARTICLES AND PROGRAMS

FEATURES

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 86 before typing in programs.

AP Apple, **Mac** Macintosh, **AT** Atari, **V** VIC-20, **64** Commodore 64, **+4** Commodore Plus/4, **16** Commodore 16, **128** Commodore 128, **P** PET/CBM, **TI** Texas Instruments, **PC** IBM PC, **PCjr** IBM PCjr, **CC** Radio Shack Color Computer.
*General interest.

TOLL FREE Subscription Order Line 800-334-0868 (in NC 919-275-9809)

COMPUTE! Publications, Inc.
One of the ABC Publishing Companies:
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019

Address all inquiries to:
P.O. Box 5406, Greensboro, NC 27403

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published monthly by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. Send subscription orders or change of address (P.O. form 3579) to **COMPUTE!** Magazine, P.O. Box 10954, Des Moines, IA 50340. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. Entire contents copyright ©1985 by COMPUTE! Publications, Inc.

ISSN 0194-357X.

Editors Notes

The subdued pallor of the personal computer section at the Summer Consumer Electronics Show was somewhat sobering. Dozens of industry vendors simply chose not to exhibit; dozens more have disappeared in the months since the last show. Noticeable in the reduced clutter of exhibitors was the increased level of professionalism and sophistication of presentation among those present. Also noticeable was the lack of industry-shaking innovation we've grown accustomed to over the last few years. Among the bright spots were our old friends at Atari, the Tramiels. They highlighted that which is best among us by promising new innovations and continued leadership at the cutting edge of truly consumer-oriented electronics. Their demonstration of an Atari/compact disc interface which allows an entire multi-volume encyclopedia to be stored and quickly retrieved from less than one-quarter of a single compact disc is truly significant. Their proposed pricing for new Atari ST systems promises hope for fall. (See the Consumer Electronics Show article elsewhere in this issue for more information.)

Commodorians are properly pushing the 128 system and reluctantly admitting the coming of the Amiga. We were shocked to discover that apparently some at Commodore still enjoy political magazine games. Several of our competitors had already received Amiga systems while Commodore public relations personnel were concurrently telling us that all magazines would be treated equally. It makes one wonder what motive Commodore might have for withholding access to the Amiga from the largest Commodore-related publisher in the industry. Ah, well. COMPUTE! always perseveres, and you may rely on us to bring you continuing and timely assessments of the new Amiga. Among our articles this month on the Consumer Electronics Show, you'll find some early information on the Amiga. It looks like a pretty impressive machine.

On this increasingly hopeful note, we'll point out that the traditionally upbeat Christmas season, while viewed with caution, is expected to be a good one for the vendors who have remained in the marketplace. It's a bit of the smaller pie and fewer slices phenomena. That same principle can perhaps be extended to the magazine publishing industry. We have a small group among our competitors whose attacks on us over the years have ebbed and flowed with the success of the various magazines they launch to compete with ours. As problems arise for whatever flagship they're currently pushing, we can detect a significant increase in the various voices they raise in criticism of us, our style, our policies, our editors, our writing. We have always chosen to remain silent in the face of these rumblings and time has always proven to be our steadfast ally. We suspect such will remain the case. In the meantime, we'll continue our efforts to always provide you with the most balanced magazines of the best quality we can publish. Thank you for your continued support.

Editor in Chief

THE BETTER LETTER BOX

Introducing EasyPlex.™ The new, easy-to-use electronic mail system from CompuServe.

Finally! Electronic Mail that's so easy to use you can start composing and sending messages the first time you get online.

Designed for various experience levels, EasyPlex has a menu mode with simple, easy-to-follow directions for beginners, and it lets experienced users save time by working in the prompt or command modes. With EasyPlex, you can compose, edit, send, file, and take advantage of sophisticated options previously available only with more expensive services.

CompuServe's EasyPlex lets friends and relatives, associations and club members communicate any time of the day or night. And small business owners, real estate professionals, insurance agents, lawyers, writers, etc. can communicate quickly and simply—either interoffice or interstate. "It's Easy." "Just Plex it!"

Best of all, EasyPlex is available to all CompuServe subscribers. And, along with EasyPlex, you get hundreds of valuable and entertaining computing options. Plus the assurance of belonging to the largest, fastest growing computer information service in the world and the premier supplier of business information to FORTUNE 500 companies.

Start communicating! To buy a CompuServe Subscription Kit, see your nearest computer dealer. To receive our informative brochure or to order direct, call or write:

CompuServe®

Information Services, P.O. Box 20212
5000 Arlington Centre Blvd., Columbus, Ohio 43220

800-848-8199

In Ohio, call 614-457-0802

An H&R Block Company

Coming In Future Issues

Hands-On Reports:
Atari 520ST
Commodore Amiga
Atari 130XE And DOS 2.5

Word Search:
Puzzle Generator For Atari, Commodore 64, 128, VIC, Plus/4, 16, Apple, IBM PC/PCjr, TI-99/4A

Chess For IBM PC/PCjr

Easy Apple Screen Editing

Commodore 64 Disk Commander

The Last Warrior Action Game For Commodore 64, Atari, Apple, IBM PC/PCjr

COMPUTE! Publications, Inc. publishes:

COMPUTE!
COMPUTE!'s GAZETTE
COMPUTE! Books
COMPUTE!'s GAZETTE DISK

Corporate office:
324 West Wendover Avenue
Suite 200
Greensboro, NC 27408 USA
Mailing address: COMPUTE!
Post Office Box 5406
Greensboro, NC 27403 USA
Telephone: 919-275-9809

Subscription Orders

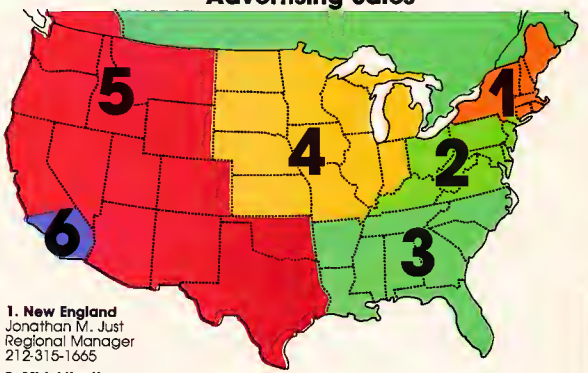
COMPUTE! Circulation Dept.
P.O. Box 914
Farmingdale, NY 11737

TOLL FREE Subscription Order Line
800-334-0868
In NC 919-275-9809

COMPUTE! Subscription Rates (12 Issue Year):

US	(one yr.)	$24
	(two yrs.)	$45
	(three yrs.)	$65
Canada and Foreign Surface Mail		$30
Foreign Air Delivery		$65

Advertising Sales



1. New England
Jonathan M. Just
Regional Manager
212-315-1665

2. Mid Atlantic
John Saval
Eastern Advertising Manager
212-315-1665
Kathy Hicks
Marsha A. Gittelman
215-646-5700
Brian S. Rogers
212-674-0238

3. Southeast & Foreign
Harry Blair
919-275-9809

4. Midwest
Gordon Benson
312-362-1821

5. Northwest/ Mountain/Texas
Phoebe Thompson
408-354-5553

6. Southwest
Ed Winchell
213-378-8361

Director of Advertising Sales
Ken Woodard

COMPUTE! Home Office 919-275-9809.

Address all advertising materials to:
Patti W. Stokes
Advertising Production Coordinator
COMPUTE! Magazine
324 West Wendover Avenue
Greensboro, NC 27408

A SPECIAL OFFER FROM
COMPUTE! BOOKS

Buy any of the new Apple titles listed below before September 15, 1985 and receive a $1.00 discount on each book. These information-packed books contain games, applications programs, buying guides, step-by-step instructions, product reviews, special tutorials for kids, and much more for your Apple II, II+, IIc, IIe, or Macintosh computer.

COMPUTE!'s Kids and the Apple
Edward H. Carlsan
$12.95
A fun and easy-ta-use baak far kids ages 10–14 interested in learning BASIC programming an their Apple computers.

COMPUTE!'s First Book of Apple
Edited
$12.95
A callectian af 35 exciting games, educatianal pragrams, hame applicatians, and graphics rautines far awners af Apple II, II+, IIe, and IIc computers.

COMPUTE!'s Easy BASIC Programs for the Apple
Brian Flynn
$14.95
A wide-ranging selectian af shart BASIC pragrams far the Apple II series af camputers, including thaught-pravoking games, hame applicatians, and business farecasting taals.

COMPUTE!'s Guide to Telecomputing on the Apple IIc
Thamas E. Enright, Jaan Nickerson, and Anne Wayman
$9.95
An infarmative and easy-ta-understand baak far the beginner which explains haw ta access bulletin baards, select telecammunicatians saftware, and use the Apple IIc ta cammunicate with ather camputers aver the phane

COMPUTE!'s Apple Games for Kids
Clark and Kathy H. Kidd
$12.95
An instant library of quality educatianal saftware ta teach and entertain the entire family.

Becoming a MacArtist
Vahé Guzelimian
$17.95
An excellent illustrated guide ta Macintash graphics, giving you direct access ta the pawer and flexibility af *MacDraw*, *MacPaint*, and *MacWrite*.

Mail in the caupon belaw with your payment by September 15, 1985 ta receive your special discount fram COMPUTE! Baaks.

To order your Apple titles, call toll-free 800-334-0868 (in NC 919-275-9809), or mail this coupon with your payment to COMPUTE! Books, P.O. Box 5058, Greensboro, NC 27403.

___ *COMPUTE!'s First Book of Apple*, 69-8 ..... **$12.95**
___ *COMPUTE!'s Kids and the Apple*, 76-0 ..... **$12.95**
___ *Becoming a MacArtist*, 80-9 ..... **$17.95**
___ *COMPUTE!'s Easy BASIC Programs for the Apple*, 88-4 .... **$14.95**
___ *COMPUTE!'s Apple Games for Kids*, 91-4 ..... **$12.95**
___ *COMPUTE!'s Guide to Telecomputing on the Apple IIc*, 98-1 **$9.95**

Subtatal $ _____
Less $1.00 per baak discaunt _____
NC residents add 4.5% sales tax _____
Add $2.00 PER BOOK far shipping _____
Tatal payment enclased _____

Please allaw 4–6 weeks for delivery. All payments must be in U.S. funds

ALL ORDERS MUST BE PREPAID
☐ Payment enclased (check ar maney arder)
☐ Charge ☐ Visa ☐ MasterCard ☐ American Express

Accaunt Na. _____ Exp. Date ____/____ (Required)

Signature _____
Name _____
Address _____
City _____
State _____ Zip Cade _____

7581151

Readers Feedback

The Editors and Readers of COMPUTE!

If you have any questions, comments, or suggestions you would like to see addressed in this column, write to "Readers' Feedback," COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Due to the volume of mail we receive, we regret that we cannot provide personal answers to technical questions.

Modular Phone Booths

I'm writing a book—not on computers, but on radio news. Like you, we make great use of the telephone for relaying material recorded on cassette. Ours is analog voice material, and the similarity of what we do to computer applications of the phone prompted me to write.

I've tried without success to interest Ma Bell and GTE in installing modular jacks on their pay phones. This would allow us to use a simple patch cord to go from a cassette recorder output without wrenching off the phone handset cover or using an acoustic coupler, which lowers quality.

Are you aware of any attempts by computer users (or manufacturers) to get direct access to phone equipment? I would imagine this would be valuable for both groups, doing away with the need for acoustic couplers, plus their extra cost and size.

I'd appreciate hearing of any efforts you're aware of on pay phone access. It may take the clout of manufacturers, computer users, and computer publications to convince these giant phone utilities to allow direct pay phone access.

F. Gifford

We haven't heard of any such lobbying efforts among computer hobbyists, but your most likely allies would be user groups that cater to portable computer owners. For instance, there's a special interest group (SIG) on the CompuServe Information Service for users of the Radio Shack TRS-80 Model 100. As active members of a commercial telecomputing network, these people are also likely to have encountered the same problems that you have. Battery-powered lap portables with built-in direct-connect modems are wonderfully convenient for traveling journalists and business people, but as you point out, the acoustic cups necessary to link them to pay phones are bulky, clumsy, and less reliable.

However, it seems unlikely that the phone companies will bend to your demands anytime soon. For one thing, portable computer users (and radio journalists) encompass a pretty small minority at this time—too small, we suspect, to justify modifying all the pay phones in the country. More importantly, handsets attached to public phones with modular jacks would be tempting targets for thieves. Anybody could unplug the handset and run off with it. Of course, pay phones could be redesigned with a conventionally attached handset and a modular jack as an accessory. Perhaps this will happen someday when portable computers are built into wristwatches and nearly everybody has one.

By the way, while you're lobbying, you might also want to target hotels and motels—we've found that many of them don't equip their phones with modular jacks, either.

Fate Of The PCjr

Being the owner of a PCjr and with the recent bad news from IBM, it seems I have to make a decision on my future with the Junior. Hopefully you can give me some insight.

1. I could sell it and then buy a PC, but that would cost a thousand or two more for a system with similar color capabilities.

2. I could move to an Apple IIc, but I would have to start all over with my software.

3. I could make the Junior as PC-compatible as possible.

I would like more help with this third choice. I have heard of two expansion chassis, one by Quadram and another by Racore. Both add a second floppy drive, clock, parallel printer port, etc. And they add a switch to change modes from PCjr to PC. The Racore also adds an optional ten-megabyte hard disk.

Could you test these add-ons? Which is better, a second floppy drive or a hard disk? Will these chassis help to secure what I've invested in the Junior, or should I bail out altogether?

Bob Hana

There's no reason to get rid of your PCjr as long as it meets your needs—and that's something only you can decide. IBM has not abandoned the PCjr; although production has been halted, IBM promises to continue supporting the computer with service and software. Since the PCjr already is fairly compatible with the PC, a wide selection of software is available and will continue to be available.

According to estimates we've seen, roughly 300,000 PCjrs have been sold. That's not a huge base compared to Commodore, TI, Apple, and Atari computers, but it's large enough to guarantee that software and expansion hardware will remain in supply in the immediate future. Still, in time, PCjr-specific products—particularly from non-IBM suppliers—may begin to dry up. So if there's anything you think your system might need, you should plan to buy while it remains available.

If you need to make your PCjr more PC-compatible, you must balance the cost of expanding the Junior against the cost of a new PC or compatible. There are several expansion modules on the market in addition to the products you mention which add more RAM, a second floppy disk drive, a realtime clock, parallel printer port, hard disk drive, and so on. Some of them allow more expansion than others and different combinations of options. See the September 1984 issue of COMPUTE!'s PC & PCjr magazine for reviews of the Tecmar jrCaptain and Legacy expansion modules.

Be aware, however, that no matter which one you pick, your PCjr won't be 100 percent PC-compatible 100 percent of the time because of some fundamental design differences. (See "PCjr Memory Compatibility," COMPUTE!, March 1985.) Usually this isn't a major concern, but you should test new software on the PCjr before buying, or at least secure return privileges in case the program doesn't work.

The question of whether a hard disk is preferable to a second floppy drive depends on your needs and your pocketbook. A hard disk is much faster and stores much more data than a floppy drive, but it

For COMMODORE and ATARI

WHEN BATTERIES INCLUDED SET OUT TO DESIGN THE VERY BEST WORD PROCESSOR FOR ATARI® COMPUTERS... THEY FOUND THEY ALREADY HAD IT.

Features all standard word processing functions PLUS

- Full Screen Editing
- Dual text windows with cut and paste between windows.
- Video preview with scrolling, lets you see exactly what will be printed, *before* it is printed
- Transfer text between documents – easily, simply
- Horizontal scrolling create wide documents with ease
- Re-arrange words, phrases, sentences, paragraphs, and columns with just a few keystrokes
- More than 40 different printer drivers – plus printer editor allows programming for any conceivable new printer
- Easy to learn, easy to use

COMPATIBLE WITH ATARI 400, 800, 600 XL, 800 XL, 1200 XL, 65 XE, 130 XE

PAPERCLIP™

Few word processors have allowed Atari users to tap the full resources of their computer ***until Atari Paper Clip****... Atari Paper Clip is an extremely powerful, fully featured word processor that will allow your Atari to operate to the limits of its potential, with an ease of operation and speed you've never thought possible.*

PAPERCLIP FOR ATARI® AND COMMODORE® OWNERS WHO WANT THE VERY BEST IN WORD PROCESSING.

BATTERIES INCLUDED

"The Energized Software Company!"

30 Mural Street
Richmond Hill, Ontario
L4B 1B5 CANADA
(416) 881-9941
Telex: 06-21-8290

17875 Sky Park North, Suite P
Irvine, California
USA 92714
(416) 881-9816
Telex: 509-139

WRITE TO US FOR FULL COLOUR CATALOGUE of our products for COMMODORE, ATARI, APPLE and IBM SYSTEMS

FOR TECHNICAL SUPPORT OR PRODUCT INFORMATION PLEASE PHONE (416) 881-9816

costs a lot more, too. One thing to keep in mind is that some commercial software is copy-protected in such a way that it requires you to boot off the floppy even if you have a hard disk.

Commodore INPUT Revisited

Your answer to Scott Mefferd's letter about suppressing the Commodore INPUT question mark (COMPUTE!, May 1985) is incorrect. It's quite easy to suppress the question mark that INPUT usually prints. Use POKE 19,64 before the INPUT command to disable it, and POKE 19,0 to bring it back. Here is an example:

```
10 POKE 19,64:INPUT"ENTER
   WORD";A$
20 POKE 19,0:PRINT
```

You must enter some value when using this method (you can't just press RETURN). Add a PRINT statement after the input, since the cursor doesn't automatically go to the next line. You can also treat the keyboard as a peripheral, reading it with an INPUT# statement as shown here:

```
10 OPEN 1,0: PRINT"ENTER
   WORD";:INPUT#1,A$
20 PRINT: CLOSE 1
```

David Tucci

A number of readers have written to suggest these methods, both of which work fine. The first method is simple and trouble-free provided you always restore things to normal with POKE 19,0. The second method takes advantage of the fact that the keyboard is just another peripheral (device number 0) as far as the computer is concerned. You can OPEN a communication channel to the keyboard and input a string with INPUT#, the same as with other peripherals.

A third method, suggested by reader Robert Kodadek, bypasses the BASIC INPUT routine and calls CHRIN directly. CHRIN is a machine language routine stored in the computer's Read Only Memory (ROM) which fetches one character from the designated input device every time it is called. Since the keyboard is the computer's default input device (unless you specify otherwise), CHRIN acts much like GET, retrieving one character at a time:

```
10 A$="": PRINT "ENTER WORD: ";
20 SYS 65487: A=PEEK(780): IF
   A<>13 THEN A$=A$ +
   CHR$(A):GOTO 20
30 PRINT: PRINT A$
```

CHRIN stores the character's ASCII value in the microprocessor's accumulator register, which is echoed at location 780 in the Commodore 64 and VIC-20. If you have a Plus/4 or Commodore 16, substitute the address 2034 for 780 in line 20. This method is a little slower than the first two because it has to compile the string one character at a time in BASIC, terminating when it detects a carriage return (CHR$(13)).

Missing Atari Memory?

When I run the memory test on my Atari 800XL, it seems to check only the first 40K of RAM. There are no red blocks anywhere on the screen, but it refuses to check the last 8K of user RAM. When I check RAMTOP with PEEK(106), it returns a value of 160. If I am not mistaken, 48K of RAM should return a value of 192. The only other symptom is an above-average amount of keyboard lockup. What's wrong here?

Dave Nessell

Either you did not disable BASIC on powerup by holding down the OPTION key or you have a cartridge installed. A cartridge or the built-in BASIC uses the top 8K of your 48K of memory. To free up this 8K of RAM, disable BASIC or remove the cartridge when running the memory test.

The keyboard lockups are probably unrelated to the results of the memory test. Instead, BASIC is most likely to blame. The first Atari BASIC cartridge suffered from a lockup bug that was supposedly fixed in revision B BASIC, the version built into the 600XL and 800XL. Unfortunately, the fix only made the problem worse. (See "INSIGHT: Atari," COMPUTE!, May and June 1985.)

Atari has finally eliminated the lockup bug for good in revision C BASIC. This version is built into the new 130XE computer and is available on cartridge for earlier machines. To obtain a cartridge, send $15 to:

Atari Corp.
Customer Relations
390 Caribbean Drive
Sunnyvale, CA 94088

Resetting The SID Chip

Does SYS 64738 completely reset the Commodore 64 to its power-up state? When I use this SYS after running a music program, and then run a game program, I can hear a faint lingering tone. This does not happen when I turn the computer off and on, then run the game program.

Bruce Snider

You've noticed a 64 "feature" that many programmers overlook. Though you might expect system reset to clear the 64's SID (Sound Interface Device) chip, all it does is turn the volume down. This is easy to demonstrate. Turn up the volume on your TV or monitor and enter the following line in direct mode (without a line number):

```
POKE 54273,20:POKE 54277,15:POKE
54278,240:POKE 54276,33:POKE
54296,15
```

Press RETURN after typing this line. The SID chip produces a continuous tone. Now type SYS 64738 and press RETURN, or press RUN/STOP-RESTORE. The volume cuts off (you may still hear a faint tone in the background). Enter POKE 54296,15 to turn up the volume again, and the tone comes back loud and clear, proving that the other SID registers retained the values you POKEd in.

SYS 64738 makes the computer jump into ROM and execute several reset routines. One of these ROM routines—called IOINIT—is supposed to reset the system for normal input/output operations (IOINIT also executes when you press RUN/STOP-RESTORE). Unfortunately, rather than putting zeros in all 25 of the SID chip's control registers (as it should do to turn off the whole chip), IOINIT just puts a zero in the volume register (location 54296). If other SID registers are still active, crosstalk *signals may leak through to the chip's output wire, producing background noise even though the SID's volume is off.*

Besides adding unwanted crosstalk, residual SID values can prevent sounds from being heard. The three low bits of location 54295 control whether any of the SID's three voices are routed through the SID filter. If any of these three bits are left on (set to 1) and the filter cutoff frequency remains at an extreme value, one or more of the voices may be distorted or inaudible.

To eliminate such problems, use the statement **FOR J=54272 TO 54296: POKE J,0:NEXT** *at the beginning of every 64 program that uses sound. You can also execute the statement in direct mode by typing it in without a line number and pressing RETURN. Incidentally, since the Commodore 128 emulates a 64 when running in 64 mode, it suffers from the same problem; however, in 128 mode RUN/STOP-RESTORE seems to clear the SID chip correctly.*

Hex Keypad For Apple MLX

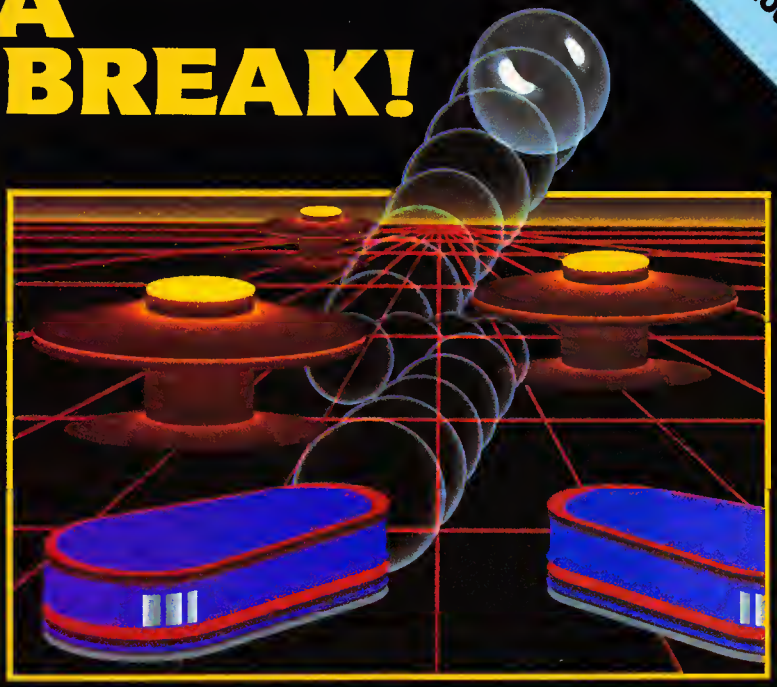
Like Larry Watkins ("Readers' Feedback," May 1985), I usually enter MLX machine language listings with one hand and follow the code with the other. Is it possible to write a program for the Apple IIc that changes the keys to a numeric keypad? I'd like to see a hexadecimal arrangement and a colon you don't have to shift.

Bill Pearson

Only two line changes are required to redefine part of the keyboard as a 16-key hexadecimal keypad for "Apple MLX" (which first appeared in the June 1985 issue and is published periodically in COMPUTE!). Replace line 410 of MLX and

TAKE A BREAK!

For Apple, Atari®, & Commodore 64™



WITH NIGHT MISSION PINBALL

You deserve the best. You've earned it. Now reward yourself with a session of Night Mission PINBALL, the most realistic and challenging arcade simulation ever conceived! ■ Stunning graphics and dazzling sound effects put Night Mission PINBALL in a class by itself. Game features: multi-ball and multi-player capabilities, ten different professionally designed levels of play, and an editor that lets you create *your own* custom modes. ■ So take a break with Night Mission PINBALL from SubLOGIC. Winner of *Electronic Games* magazine's 1983 Arcade Award for Best Computer Audio/Visual Effects.

See your dealer . . .

or write or call for more information.

Order Line: 800 / 637-4983

subLOGIC Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

add line 415 as shown here;

```
410 FOR I=1 TO 17: IF K < > ASC(
    MID$( "M,./JKL;UIOP7890",
    I, 1)) THEN NEXT: GOTO 420
415 A$ = L$ + MID$( "012345678
    90ABCDEF", I, 1) + R$: P = P + 1
```

Once these changes are made, Apple MLX accepts 7-8-9-0 for C-D-E-F, U-I-O-P for 8-9-A-B, J-K-L-; for 4-5-6-7 and M-,-.-/ for 0-1-2-3. You can even put stick-on numbers on the front of the redefined keys. Since you don't need to type colons in Apple MLX listings, the colon key has been left alone.

Better Atari Color Combinations

Please tell me the proper POKE or SETCOLOR command to make my Atari 800XL's text blue on darkest blue, blue on black, white on black, or any other combinations that might be easier on one's eyes. Will leaving the computer in these modes for long periods of time damage anything? Is there any way to make DOS 3.0 work in these altered text colors?

Jeb Branham

It's quite easy to change the Atari screen colors. The statement SETCOLOR 1, color,brightness *sets the brightness level of text. The* color *value is irrelevant, since text is always the same color as the background (simply a different shade). The* brightness *value must be an even number from 0 (darkest) to 14 (brightest). Use* SETCOLOR 2,color,brightness *to control the background color. The* color *value can be any number from 0–15, and the* brightness *can be any even number from 0–14. You can also set the border color with* SETCOLOR 4,color,brightness.

For instance, the statements **SETCOLOR 1,0,10: SETCOLOR 2,9,0** *produce light blue on dark blue. Blue text on a black background is not possible, since both screen and character color must be the same color. However, some shades of the same color look like different colors. For example, bright red-orange looks like yellow, and white is actually "bright black." Thus,* **SETCOLOR 1,0,10: SETCOLOR 2,0,0** *gives you white text on a black screen.*

Many people find it easier to read black text on a white background, since this combination simulates the appearance of type on paper. Use **SETCOLOR 1,0,2: SETCOLOR 2,0,10**. *You may have to fiddle with the brightness numbers to get the contrast right. Unfortunately, these color changes are transient. The normal screen colors return when you press SYSTEM RESET, change graphics modes, or go to DOS. To change the screen colors of the DOS menu, you'd have to disassemble DOS to find the instruction which sets the colors and then alter the instruction yourself.*

No color combination will damage your TV or monitor unless you leave very bright text on the screen for a significant period of time (such as overnight). Atari computers have a built-in protection feature against burn-in: If you don't press any keys for about nine minutes, the computer automatically enters attract mode, *in which the screen colors continually cycle at 50 percent brightness until you press a key.*

Programming The VIC/64 User Port

I have built a breadboard system and interface to the VIC-20 user port, but am having trouble with programs to make use of it. Could you give me more information on how to program the user port?

John W. Farrow, Sr.

The user port, located on the back of the computer on the left side, gives you direct access to the computer and allows control of external parallel and RS-232 serial devices. Access to the user port is through the VIA (Versatile Interface Adapter) chips on the VIC, and the CIA (Complex Interface Adapter) chips on the 64.

Communications with RS-232 serial devices like modems are provided for in the computers' operating system via device 2, so we assume your homebrew interface makes use of the user port's eight-bit parallel data port. The parallel port can be controlled directly from BASIC with PEEK and POKE commands. When the port is being used for input, the address (37136 for the VIC, 56577 for the 64) is PEEKed. When the port is used for output, the address is POKEd.

Before data can be exchanged through the port, the function of the eight data lines must be specified by setting the data direction register for the user port (37138 for the VIC, 56579 for the 64). Each of the eight bits at this address controls the direction of data flow for the corresponding bit of the user port. When a bit in this register is set to 0, the corresponding bit in the user port is used for input. Setting a bit in the data direction register to 1 indicates that the user port bit will be used for output. Pressing RUN/STOP–RESTORE or powering up initializes all bits in the direction register to 0, setting all lines of the port for input. POKEing a value of 255 into the register will set all lines for output. Any combination of input and output lines can be specified by POKEing the value for the desired pattern of 1's and 0's into the data direction register.

Once the data direction register is set up, the desired lines of the user port can be read from or written to by PEEKing or POKEing the data register. If a line is selected for data input, the corresponding bit in the data register will hold a 0 if the line is at its low state (0 volts) and a 1 if the line is at its high state (at least 2.4—but not more than 5—volts). If the port is set for output, setting a bit in the data register to 0 causes the corresponding line on the port to be set to its low state, 0 volts. Setting a bit to 1 causes the voltage on the line to rise to its high state (usually about +5 volts). For example, the following statements set all eight lines of the VIC's user port for output, then present a high (+5V) state on each line:

```
10 POKE 37138,255
20 POKE 37136,255
```

Since applying improper voltages to the lines of the user port can damage the VIA and CIA chips—rendering your computer useless—we recommend that you use caution when experimenting with the port. If you're unfamiliar with the basics of electronics, you should connect only circuits designed by knowledgeable technicians.

For more information, and a simple peripheral device which can be controlled by the user port, refer to Chapter 5 in COMPUTE!'s First Book of Commodore 64. *Additional information can also be obtained from* Mapping the VIC, Mapping the Commodore 64, *and* Programming the VIC, *from COMPUTE! Books.* ©

CLOSE ENCOUNTERS OF THE FANTASY KIND

QUESTRON.™ *A role-playing game so spell-binding, it suspends reality and conjures up an enchanted world whose only hope against the forces of evil is you.*

GEMSTONE WARRIOR.™ *A lightning-fast action strategy game that will leave you breathless with excitement as it takes you through the labyrinths of the Netherworld on a noble quest.*

Open them up and step through the gateway to your fantasies come true.
At your local computer/software or game store today.

QUESTRON™ is available on 48K diskette for APPLE®II with Applesoft ROM, Apple II+, IIe, and IIc ($49.95). On 40K diskette for ATARI® home computers ($49.95). On 64K diskette for COMMODORE 64™ ($39.95).

GEMSTONE WARRIOR™ is available on 48K diskette for APPLE®II with Applesoft ROM, Apple II+, IIe, and IIc ($34.95). Also on 64K diskette for COMMODORE 64™ ($34.95).

If there are no convenient stores near you, VISA & M/C holders can order direct by calling 800-227-1617, ext. 335 (toll free). In California, 800-772-3545, ext. 335. Please specify computer format and add $2.00 for shipping and handling.

To order by mail, send your check to: STRATEGIC SIMULATIONS INC, 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. (California residents, add 7% sales tax.) All our games carry a "14-day satisfaction or your money back" guarantee.

WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.

Report From The Summer Consumer Electronics Show

Tom R. Halfhill, Editor

Considerably less frantic than past Consumer Electronic Shows—at least in the computer section—this summer's CES nevertheless showcased some groundbreaking new products. Foremost was Atari's announcement of a mass storage device that may bring optical memory into homes, schools, and businesses by early 1986.

It would border on the outrageous to describe any Consumer Electronics Show as "quiet"—considering that 80,000 to 100,000 retailers, wholesalers, middlemen, and journalists spend four days jamming their way into convention halls for what is billed as the world's largest industry trade show.

Still, something was noticeably different about this June's CES in Chicago. The annual noise which emanates from the personal computer section in the McCormick West building had dissipated to a muffled roar.

Only two U.S. computer manufacturers were in attendance: Commodore and Atari. IBM and Apple, as is their custom, skipped the show. Atari, which a few weeks earlier had announced it was pulling out of CES, was enticed back by the show management but occupied a couple of meeting rooms instead of its extravagant exhibit of days past. And the lower level of McCormick West, once the exclusive domain of a hundred computer software companies, now was half-filled with videotape exhibits and purveyors of video porn. Rarely have the effects of the much-publicized industry shakeout been so apparent.

On the bright side, the mood was just slightly more optimistic as both Atari and Commodore moved closer to shipping actual production models of their latest personal computers. In fact, as the show opened, Atari said it had delivered the first 5,000 of its new 520STs to Canada and Europe and was expecting large-volume shipments to the U.S. by July 8. Commodore said it was only weeks away from shipping the Commodore 128, and was gearing up for a press conference in late July to officially announce its eagerly anticipated Amiga computer (see the accompanying article, "A Tantalizing Peek At The Amiga").

Several software companies announced new products for the Commodore 128, Amiga, and Atari ST series, although most seem to be cautiously hanging back until they see how the machines fare.

On the hardware front, Atari announced a mass-market version of the ST, Commodore exhibited a dual disk drive for the Commodore 128, and a British company announced it would export a 128K computer to the U.S. But the most interesting news from the show was a revolutionary new peripheral displayed for the first time by Atari: A very fast mass storage device that uses high-density optical discs. It was a dramatic demonstration that the upcoming generation of personal computers will place much of the power of a mainframe computer on a desktop. It also showed that the personal computer industry is not only far from finished, but is just getting started.

Atari's big announcement was the CD-ROM, which stands for *Compact Disc-Read Only Memory*. If you've never heard of a CD-ROM before, prepare to read dozens of articles about it in coming months, because within two years CD-ROMs will probably be everywhere. (See the accompanying article, "Monster Memory.")

Atari hooked up a working prototype of a CD-ROM to a 520ST on the second morning of the show, and the Atari exhibit was packed for the next three days. Although the hardware looked rough, the software appeared to be nearly complete. In fact, Atari hopes to have the product ready for sale by late this year or early 1986 for about $500.

The software that allows the CD-ROM to work with the 520ST is being developed by an optical typesetting company, Activenture, Inc. of Pacific Grove, California. Activenture has placed an entire 23-volume, nine-million-word encyclopedia plus index on a single optical disc—and the disc is still three-quarters empty. The system is so efficient that any entry can be looked up in a matter of seconds.

When the CD-ROM is introduced, Atari and Activenture hope to have other databases available, too. Some examples might be additional encyclopedias, legal and medical references, cookbooks, phone books, and the whole Library of Congress card catalog. In

fact, for reference works, the CD-ROM may be a significant step toward the paperless information age.

"The ink-on-paper business was fairly restrictive," says Tom Rolander, vice president for engineering of Activenture. "The only way you could look at information was in the way the original publisher had organized and presented it while laying it down on paper. When using reference material—which is why we're starting with encyclopedias and things like that—the degree of usefulness is based on how well we can find the information within that material. In other words, how good its indexing system is. What we have here, by connecting the computer with the CD-ROM, is the ultimate indexing tool. We'll know every reference there is to everything that's in the database."

The paperless information age, however, is encountering some resistance by those whose business depends on putting information on paper—traditional publishers. For example, Atari and Activenture had to delay announcing the name of the encyclopedia on their CD-ROM disc because the publisher didn't want any publicity until the final contract was signed. Rolander says all the publishers will probably fall in line once the first one does, but that many publishers are wary of the new technology.

"They can charge $1,000 for a shelf-full of books, but how much will people be willing to pay for the same information on one of these?" asks Rolander, spinning a disc on his finger. "To consumers, it doesn't look like they're getting as much. Will people pay $500? Or $150? Or $50? We don't know yet. This may finally determine the true value of information."

Besides the CD-ROM, Atari announced two new variations of its 520ST, the 260ST and 260STD. They're identical to the 520ST except for three features: 256K RAM instead of 512K; a built-in RF modulator so they can be plugged into an ordinary TV set as well as color or monochrome monitors; and the operating system and GEM (Graphics Environment Manager) in ROM instead of in RAM. Also, the 260STD has a built-in 3½-inch floppy disk drive. They are scheduled for release this fall for $399 and $499, respectively.

Atari says it plans to sell the 260ST and 260STD through mass-market channels, such as discount stores, while restricting the 520ST to specialty dealers, such as computer shops. This is a reversal of Atari's original plan to sell the 520ST through all types of outlets.

Atari also has been wavering back and forth on whether the operating system and GEM will ever be in ROM on the 520ST. When the 520ST was first announced in January, Atari said all the system software would be in ROM. But delays in debugging the operating system prompted Atari to release it on disk with initial shipments of the computer. Then there were conflicting statements about whether early 520ST owners would be able to upgrade to a ROM-based operating system later. (A RAM-based operating system takes longer to boot up, but is easier to revise; a ROM-based system boots up instantly, but can be upgraded only by replacing the ROM chips.)

At CES, Atari President Sam Tramiel told COMPUTE! that the operating system will be transferred to ROM for the 260ST/260STD, and then made available at "minimal cost" for 520ST owners with RAM-based systems. "These 18 guys back in Sunnyvale [at Atari's software department] are right now crunching the code to get it into the ROM size [192K]," said Tramiel. "TOS [Tramiel Operating System] now I think is 205K, or something like that. We feel it's not a big problem, but we've got to get it done fast."

Commodore was relatively idle at this CES. Its only new hardware announcements for the U.S. market were a dual floppy disk drive for the Commodore 128 and a dot matrix printer. Interestingly, before CES started, Commodore intended to show a hard disk drive for the 64 and 128, but pulled the product at the last minute for unknown reasons. Commodore also pulled the LCD portable lap computer first shown at the January CES; reportedly, the machine has been postponed while Commodore concentrates on bringing the 128 and Amiga to market.

The new 1572 dual disk drive combines two 1571 drives in a slimline case designed to sit atop the Commodore 128. It has the same multimode capabilities as the 1571 (Commodore 64, Commodore 128, and CP/M formats). Commodore says it should be available this summer, but no price was announced.

The new MPS 1000 is a multimode dot matrix printer. In draft mode, it prints at 100 characters per second (cps); in near-letter quality mode, it prints sharper characters at 16 cps; and in graphics mode, it has a density of 50 to 240 dots per inch. It can also print in widths ranging from 80 to 160 columns. It's compatible with the Commodore 128, 64, and many other personal computers. Like the 1572 disk drive, it's scheduled to be available this summer, but no price was announced.

Two products exhibited at the Winter CES were firmed up at this show. The Commodore 1670 direct-connect modem, which transfers data at 1200 bits per second, will sell for around $200 and has auto dial, auto answer, auto mode selection, and auto speed switching from 300 to 1200 bps. It works with the 128, 64, SX-64, Plus/4, and VIC-20. And the Commodore two-button mouse controller first seen in January will sell for $49.95 and should be available immediately. It works with the 128, 64, and VIC.

Commodore also showed four interesting computers for foreign markets, but apparently they won't be available in the U.S. in the near future. The Commodore 128D Integral Personal Computer is a variation on the 128 that separates the keyboard from the system unit and includes a built-in disk drive. Commodore says it will be available in Europe late this year. The PC10 and PC20 are IBM-compatible computers recently introduced in Europe. The PC10 has 256K RAM and two 360K floppy disk drives; the PC20 has 512K RAM, one floppy drive, and a ten-megabyte hard disk. And finally, the Commodore 900 Business Computer is a multitasking, multiuser workstation that uses a

Monster Memory

The CD-ROM, an acronym for *Compact Disc-Read Only Memory*, is a compact disc audio player which has been slightly modified for general-purpose data storage and interfaced to a computer.

Compact disc players are the latest rage among audiophiles. Up to 75 minutes of digitally encoded music can be stored in the form of microscopic pits on a 4.7-inch rigid plastic disc. Inserted in a special player, the disc spins at 300 r.p.m. while a miniature laser reads the pits. The data is decoded by a microprocessor, then converted into standard audio signals which are fed into the auxiliary input or tape monitor jacks on a stereo receiver. The result is exceptionally pure music of unprecedented dynamic range and frequency response, free of surface noise and tape hiss. Furthermore, since the disc is read by a laser, not a diamond stylus, compact discs last virtually forever with no deterioration. They can also tolerate rougher handling than ordinary records and tapes.

But music isn't the only thing a compact disc can store. *Any* type of information can be digitized and recorded on a disc. That includes text, graphics, and computer programs. And the capacity is enormous: A single compact disc stores about *550 megabytes*. A megabyte equals 1,024K, so that's roughly equivalent to 1,564 floppy disks on an IBM PC, 3,520 disks on a Commodore 1541 drive, 4,022 disks on an Apple II, 4,469 enhanced-density disks on an Atari, or 6,400 single-density Atari disks. They're cheap, too: compact discs can be mass-produced at a manufacturing cost of a few dollars each (audio discs currently retail for about $15). Because audio CDs and CD players are already in mass production, CD-ROMs can debut at affordable prices.

A compact disc is a read-only storage medium, so you can't record data on it yourself. But CDs are ideal for storing large databases that don't have to be updated often. At CES, Atari demonstrated a sample disc that contained a 23-volume, nine-million-word encyclopedia. The encyclopedia was transferred to the CD from magnetic tape, where it was stored in punchcard format—the equivalent of 976,000 punchcards. Yet, it fits on one-quarter of the space of a single CD.

To think of a CD simply as an efficient way to store mass amounts of information is to miss the point, however. Like a floppy disk drive, a CD player is a *random-access* device; it can seek and retrieve any piece of data on the disc in a few seconds without hunting through the information sequentially. Therefore, a CD-ROM can find the slightest, most obscure fact in a massive database in less time than it takes you to pull a book off a shelf and flip it open to the index.

Here's an example: Let's say you're a student researching a report on Thomas Jefferson. On the Atari CD-ROM, there are two ways to approach the task.

The first way is very similar to the usual method of looking up something in an encyclopedia. First, you boot up the CD-ROM on the Atari 520ST. This takes only a few seconds. A graphics display on the screen shows a bookshelf with a 23-volume encyclopedia. By rolling the ST's mouse controller, you move the screen pointer to the "J" volume and then click the mouse button. This brings up another screen with a graphics picture of the book you selected, opened to several alphabetical tab entries. You move the pointer to the tab which would include *Jefferson*—for instance, *Japan to Jet*. Another click calls up a screen showing all the article titles within that section. When you move the pointer to the title *Jefferson, Thomas* and click the button again, the computer loads the article (and several preceding and following articles, as well) from the CD into memory. It takes less than four seconds for the CD-ROM to fill the 520ST's entire 512K RAM.

Now you can read the article on the screen, scrolling or flipping pages by clicking the mouse. This method of looking up subjects is recommended for those who aren't familiar with computers, because it requires almost no computer knowledge.

The second method takes greater advantage of the computer's power. Instead of looking up the subject alphabetically by yanking a graphics image of a book off a shelf, you pull down a menu and select the *search screen*. This screen presents a number of options; to keep it simple, you can just type *Thomas Jefferson* at the prompt and ask for a general search. In about four seconds, the computer reports how many times the phrase *Thomas Jefferson* appears in the encyclopedia. You can flip to the first occurrence by clicking the mouse. Again, in less than four seconds, the computer loads the article from the CD into memory and even highlights your search phrase within the text. You can flip to subsequent occurrences merely by clicking the mouse button.

What makes this technique so powerful is that the computer will find references to Thomas Jefferson in articles that may never have been checked using the old-fashioned method. The student may learn that Jefferson was not just a politician, but also an inventor, architect, and connoisseur of wines. Looking up the same references in even the most thoroughly cross-indexed paper encyclopedia would be much more time-consuming.

When the Atari CD-ROM software is finished, it will allow two-dimensional searches, too. You could look up every article that contains references to Thomas Jefferson *and* Thomas Paine, or Thomas

Get the jump on the weather-man by accurately forecasting the local weather yourself!

A scientifically proven way to develop an awesome memory.

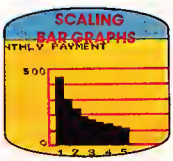
You are trapped in a five-story, 125-room structure made entirely of ice. Find the exit before you freeze!

Take control of your personal finances in less than one hour a month.

The beautiful princess is held captive by deadly dragons. Only a knight in shining armor can save her now!

Cut your energy costs by monitoring your phone, electric and gas bills.

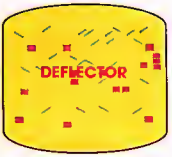
Computerize car maintenance to improve auto performance, economy and resale value.



Create multi-colored bar graphs with a surprisingly small amount of memory.

A time-saving organizer for coupons, receipts and more.

School-age and pre-school children are rewarded for right answers, corrected on their wrong ones.



A real brainflexer. Deflect random balls into targets on a constantly changing playfield.

A fun way to dramatically increase typing speed and accuracy.

Get up to 30 new programs and games for less than 15 cents each— every month in COMPUTE!

Every month, COMPUTE! readers enjoy up to 30 brand-new, ready-to-run computer programs, even arcade-quality games.

And when you subscribe to COMPUTE!, you'll get them all for less than 15 cents each!

You'll find programs to help you conserve time, energy and money. Programs like Cash Flow Manager, Retirement Planner, Coupon Filer, Dynamic Bookkeeping.

You'll enjoy games like Air Defense, Boggler, Slalom, and High Speed Mazer.

Your children will find learning fast and fun with First Math, Guess That Animal, and Mystery Spell.

Looking for a challenge? You can write your own games. Customize BASIC programs. Even make beautiful computer music and pictures.

It's all in COMPUTE!. All ready to type in and run on your Atari, Apple, Commodore, PET/CBM, TI-99/4A, Radio Shack Color Computer, IBM PC or IBM PCjr.

What's more, you get information-packed articles, product reviews, ideas and advice that add power and excitement to all your home computing.

And when it's time to shop for peripherals or hardware, check COMPUTE! first. Our product evaluations can save you money and costly mistakes. We'll even help you decide what to buy: Dot-matrix or daisy-wheel printer? Tape storage or disk drive? What about modems? Memory expansion kits? What's new in joysticks, paddles and track balls?

Order now! Mail the postpaid card attached to this ad and start receiving every issue of COMPUTE!.

For Faster Service Call Toll-Free 1-800-334-0868

COMPUTE! P.O. Box 914, Farmingdale, NY 11737

Jefferson *or* George Washington. Other options let you limit the search for occurrences in adjacent words, single paragraphs, and word groupings of various sizes.

If you're an experienced programmer, you might be puzzled by the search times described above. Even in superfast machine language on the 520ST's 68000 microprocessor, how can the computer search through nine million words in less than four seconds?

The answer is that the computer can't. Instead, it refers to an extremely sophisticated index of search tables on the CD.

The search tables were compiled with a VAX minicomputer by Activenture, Inc., the company which is developing the CD-ROM software for Atari. First, the VAX built a dictionary by identifying every unique word in the encyclopedia—more than 140,000 words. Then it compiled search tables which cross-reference the dictionary to every occurrence of each word in the encyclopedia. In conventional book form, the finished index would occupy about 20 volumes. That means the index is nearly as long as the encyclopedia itself.

When you request a search, the 520ST simply consults the index of search tables on the disc and rapidly locates each occurrence of the search phrase. In effect, the searching has already been done for it by the VAX.

Retrieving the information is very fast, too, though not quite as fast as with hard disks. Because CDs were designed for storing music, which is played sequentially, their random-access capabilities are not as good as they could be. Still, they're much faster than most floppy disk drives. Data is stored on a CD in 270,000 records containing 2,048 bytes each; the average access time for a record is about one second. The greatest access time—which happens when the head must move from an extreme outside track to an inside track or vice versa—is three to four seconds. To keep this from happening very often, data is recorded on a CD on the inside tracks first, moving outward. This takes advantage of the CD-ROM's very rapid track-to-track access time. To read a nearby track, the player merely tilts a tiny mirror to refocus the laser rather than repositioning the entire head. (Each track of microscopic pits is only two microns—two *millionths* of a meter—wide).

Once the information is located, the CD-ROM feeds it to the computer at a rate of 75 records (150K) per second. To make sure the data arrives without errors, the CD-ROM's error-correction capabilities have been improved over that of a regular audio CD player. An audio player can be expected to pass one bad bit for every 10,000 bits—inaudible when listening to Beethoven or the Beatles, but not nearly good enough for computer storage. So CD-ROMs employ an error-checking and correction scheme that allocates an additional 288 bytes for each 2,048-byte record. That much overhead—more than 14 percent—would be wasteful on a floppy or hard disk, but CDs have room to burn. The improved error rate on a CD-ROM is at least as good as with a hard disk: about one bit error for every 1,000,000,000,000 to 1,000,000,000,000,000 bits (one trillion to one quadrillion).

Report From Summer CES Continues

Unix-compatible operating system.

Although some of these higher-end computers will be available in Canada, Commodore announced no plans at present to market them in the U.S.

Another new personal computer was announced at CES by a British company, Amstrad. Already available in Europe, where several hundred thousand units have been sold, the Amstrad CPC6128 is scheduled to be shipped to the U.S. later this year.

The U.S. version of the Amstrad has an 8-bit Z80A microprocessor, 128K RAM, a built-in 3-inch disk drive, CP/M compatibility, BASIC and Logo, an expansion interface, joystick port, and stereo sound output. It comes packaged in two configurations. One includes a green-screen monitor and *WordStar* word processor for $699, and the other has an RGB color monitor, Amstrad's own word processor, and some entertainment software for $799.

Briefly, here are some other highlights of the Summer CES:

• Abacus Software of Grand Rapids, Michigan announced *Super C*, a C compiler for the Commodore 64 and 128. It has a full-screen editor with horizontal and vertical scrolling and is compatible with most other versions of C. Source files up to 41K long can be created.

• Commodore is releasing several titles for the 128, including *Jane 2.0*, an icon-based integrated package with a word processor, spreadsheet, and filing manager, all of which can be manipulated with the mouse controller; *Micro Illustrator*, a graphics drawing program formerly available for the 64, which takes advantage of the 128's extra memory and other features; and the *Perfect* series for the CP/M operating system, consisting of *Perfect Writer*, *Perfect Calc*, and *Perfect Filer*. All three work in the 80-column mode, are capable of sharing files, and have pop-up menus, split screens, and automatic formatting for printouts.

• Epyx, Inc. of Sunnyvale, California is releasing *Winter Games*, a sequel to *Summer Games*, for the 64, Apple, and Macintosh; two new LucasFilm games for the Atari, *The Eidolon* and *Koronis Rift*; *The World's Greatest Football Game* for the 64, Apple, and Atari; and the *Temple of Apshai Trilogy* for the 64, Apple, Atari, Macintosh, and IBM. (Prices will range from $19 to $35.)

• Batteries Included of Richmond Hill, Ontario is releasing its *PaperClip* word processor for the Apple ($89.95) and Commodore 128 ($119.95 with *SpellPak*); the BI-80, an 80-column video adapter on a cartridge for the Atari XL and XE series ($79.95); an 80-column version of Atari *PaperClip* for the BI-80 ($59.95); new versions of *HomePak*—a combination word processor, filer, and terminal program—for the IBM PC/PCjr, Commodore 128, Apple, and Atari ST ($49.95); an IBM version of *The Consultant*

For APPLE® and COMMODORE 64™

When sorcery ruled, and trolls and minotaurs still walked this Earth, a party of six Intrepid adventurers set out to find the Nine Rings and use them to destroy the Dark Lord. This is your quest. This is your...

PHANTASIE™

A MULTIPLE-CHARACTER ROLE-PLAYING ODYSSEY

Look for this new, exciting game at your local computer/software or game store today!

If there are no convenient stores near you, VISA & Mastercard holders can order direct by calling 800-227-1617, ext. 335 (toll free). In California, call 800-772-3545, ext. 335.

To order by mail, send your check to: STRATEGIC SIMULATIONS, INC., 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. Please add $2.00 for shipping and handling (California residents, add 7% sales tax). All our games carry a "14-day satisfaction or your money back" guarantee.

WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.

PHANTASIE is $39.95.

5¼" diskette for Apple® with Applesoft in ROM, II, II+, IIe, IIc.
Also on diskette for Commodore 64™.

Apple and Commodore 64 are trademarks of Apple Computer Inc. and Commodore Electronics Ltd., respectively.

SSI

STRATEGIC SIMULATIONS, INC.

An attack by killer bees in the colorful countryside.

The wave of deadly monsters continues as dwarfish Kobolds threaten your party.

One of your comrades is injured by Ghouls during a search of the dungeons.

A sample composition of your party showing race, class and status.

A Tantalizing Peek At The Amiga

At this writing (early June), Commodore is scheduled to unveil its long-awaited Amiga Lorraine computer at a press conference in New York in late July. Although the machine was substantially ready in June, Commodore evidently kept it off the show floor at CES to avoid stealing attention away from the Commodore 128, which was due to begin appearing on store shelves within weeks. Nevertheless, despite unprecedented secrecy and security, more and more details about the Amiga leaked out at CES press parties. Also, COMPUTE! arranged a peek at the Amiga through a helpful source. We learned that even more capabilities have been added since our first look at the computer a year ago at the June 1984 CES (see "Software Power! The Summer Consumer Electronics Show," August 1984).

The Amiga's final configuration was still subject to change before its official introduction, but here's what it will probably include:

• Motorola 68000 microprocessor for the central processing unit. This is the same 16/32-bit chip found in the Apple Macintosh and Atari ST series. It can address up to 16 megabytes of memory (1 megabyte = 1024K).

• 256K of RAM (Random Access Memory), expandable to at least 512K on the system unit and more externally.

• 192K of ROM (Read Only Memory) containing *Intuition*, a Macintosh-like operating system with pull-down menus, icons, hierarchical disk directories, multitasking, and mouse support. Unlike the Macintosh's operating system, however, *Intuition* can be manipulated with keyboard controls as well as the mouse. The keyboard, which is attached to the Amiga system unit with a coiled cord, includes cursor keys and a numeric keypad.

• True multitasking. Thanks to the high-speed 68000 and a number of dedicated chips for input/output and other vital functions, the Amiga can run several programs simultaneously with no apparent sluggishness, even while simultaneously accessing the disk drive. By opening screen windows of various sizes, you can watch all the programs running at once. This is an especially useful feature for business applications—you could simultaneously work with a word processor, terminal program, database manager, and spreadsheet without stopping one program to start another.

• A built-in double-sided 3½-inch disk drive that stores about 800K per disk. These are the same hard-shell microfloppy disks used by the Macintosh and Atari ST series. Up to three external drives can be added to the Amiga by daisy-chaining.

• Custom chips for graphics and animation. Maximum screen resolution is 640 × 400 pixels (screen dots), with additional graphics modes of 640 × 200 and 320 × 200. Eighty-column text is standard, but the display is adjustable to narrower widths for greater readability on low-resolution screens.

• Composite video/mixer input that allows you to feed standard video signals into the Amiga, display them on the screen, and then superimpose the Amiga's text and graphics. The external video source could be a video camera, videocassette recorder, videodisc player, TV receiver, or even another computer. The potential of this feature is exciting: games and educational programs with superrealistic backgrounds, titles for home videotapes, and so on. Furthermore, a relatively low-cost peripheral called a *frame grabber* lets the Amiga digitize the incoming video signal so it can be manipulated with graphics utilities, stored on disk, and even dumped to a graphics printer.

• A palette of 4,096 colors, more than any other general-purpose personal computer on the market. Up to 32 of these colors can be displayed at once without special tricks.

• Video outputs for TV sets, composite color and monochrome monitors, and high-resolution RGB color monitors. Reportedly, the RGB output is analog, like the Atari ST's, so it's not compatible with RGB monitors designed for the Commodore 128 and IBM PC computers. Analog RGB allows more color intensity levels than IBM-type RGB.

• An expansion port that includes every line on the system bus for almost unlimited expansion capabilities. This could include coprocessors, such as an 8088/8086 board for IBM compatibility. Reportedly, the Amiga will be an "open system." That is, to help independent hardware and software designers access the full power of the computer, Commodore is said to be preparing extremely detailed documentation on all aspects of the Amiga for general release. One insider who has seen the preliminary documentation says it's so complete you could almost build the system from scratch using the information it contains.

• Seven-level direct memory access (DMA) controller. Along with the Amiga's many dedicated chips, this lets the machine perform several tasks simultaneously with

ONLINE IN THE U.S.A.

Be anything you want on American PEOPLE/LINK, where people from across the country get together every night.

They are people with interests in sports, movies, music, games, and just meeting other people . . . people like you. Whatever your interests or fantasies, you'll find someone to share them with on PEOPLE/LINK.

See for yourself why so many men and women are getting together and partying online with us. Not screen after screen of boring data . . . just lots of fun and friendly people.

And our low, low rates won't turn your online dreams into nightmarish monthly charges. To discover our affordable videotex entertainment, call or write us today.

PEOPLE/LINK . . . the network that says "Thank You."

AMERICAN PEOPLE LINK

American PEOPLE/LINK
Arlington Ridge Office Center
3215 N. Frontage Road – Suite 1505
Arlington Heights, IL 60004

1-800-524-0100 (Toll Free) 1-312-870-5200 (Illinois)

no apparent slowdown. It also simplifies things for programmers. An independent software developer who attended a recent Amiga symposium sponsored by Commodore in Monterey, California, told COMPUTE! that systems-level programming on the computer is so easy that "it's more like parameter-passing than programming. You just decide what you want to do—pull data off the disk, whatever—and pass the appropriate parameters to the operating system, and the OS passes them along to the DMA controller, and everything happens during the 68000's off-phase cycle so the machine doesn't slow down at all." (The programmer's symposium, incidentally, was open only to independent developers and was sealed off by armed guards. Those who attended had to sign a strict nondisclosure agreement.)

• Built-in printer drivers to dump the Amiga's bitmapped screens onto graphics printers, including color and inkjet printers.

• Four-voice sound chip with stereo output. The Amiga's custom sound chip routes two voices to each stereo channel for high-fidelity reproduction through external stereo systems. In addition, the sound chip is the most advanced in any personal computer, surpassing even the Commodore 64's SID chip. The Amiga can closely simulate a wide variety of musical instruments, and at least a dozen instrument sounds are built in—such as guitar, pipe organ, cymbals, drums, piano, and violin. Sound envelopes (attack-decay-sustain-release) can be modified simply by pulling down a menu and making selections with the pointer.

• Digital sound sampling. At this point, it's not clear whether this feature will be standard or optional. Even if it's optional, however, insiders say it will cost much less than anything similar now on the market (the least expensive high fidelity samplers now cost around $2,000). Digital sound sampling lets you feed sound from an external source into a computer, convert it to digital format, and then play it back, modify the waveforms, or store it on disk. With this capability, programmers and musicians no longer have to spend hours trying to simulate a musical instrument or sound effect—they can just feed the sound directly into the computer from a record player, tape deck, microphone, or instrument, and then manipulate it at will. In fact, some Amiga software developers are taking this approach to cut down development time.

• Built-in speech synthesis. The operating system includes a speech program with text-to-speech conversion. Since this is a standard feature that requires no extra hardware, you can expect many programs to take advantage of the Amiga's speech capabilities—including programs written by home users. COMPUTE! has heard the Amiga talk, and its pronunciation was more understandable than most speech synthesizers now available for personal computers.

• BASIC programming language. Unlike the Macintosh, the Amiga will come with a language for those who want to write their own programs, but it's unclear at this point if BASIC will be built into ROM or loaded from disk into RAM. Pascal and C compilers will be available soon after the computer's release. A 68000 assembler also will be offered, but it may not be as popular as assemblers on the current generation of personal computers—the compilers are so fast that few programmers are using machine language. One software developer says he has yet to see a single program written for the Amiga in machine language.

• Hard disk option. Although Commodore is said to have no immediate plans for a hard disk drive, an independent company known for its IBM peripherals is rumored to be preparing a hard disk with 10 to 20 megabytes of storage. This would probably interface to the expansion port.

If you take the word of those who have worked with the Amiga, it is the most powerful personal computer ever assembled. In terms of performance, they say it's more than a match for any business-oriented machine now on the market, and has the advanced graphics and sound features demanded by home users as well. But how much will it cost?

At the Monterey symposium, Commodore reportedly told developers that the Amiga would be sold with an RGB monitor and some software for about $2,000. Since then, however, Commodore has encountered some negative reaction to that figure. Many observers think a lower price is necessary to dramatically undercut the Macintosh and IBM PC, and also to compete with the Atari ST series for a larger share of the intermediate-priced market. At CES, rumor had the price ranging anywhere from $1,300 to $1,900, possibly without an RGB monitor.

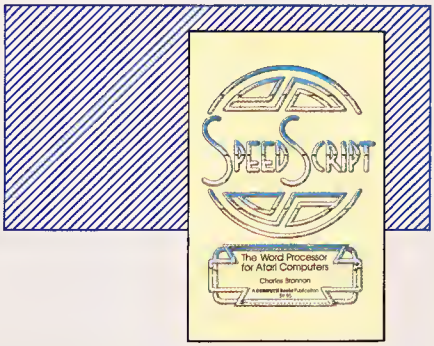
Report From Summer CES Continues . . .

database manager ($99.95); *Battery-Pak*, a seven-function desktop accessory for the Macintosh ($49.95); and *Literature Challenge: Introduction to Shakespeare*, an educational program for the Apple, Commodore 64, and IBM PC ($29.95). In addition, Batteries Included announced a new line of integrated programs for the Atari ST, Commodore Amiga, and MS-DOS computers with GEM. Called the IS line, all the programs will have Macintosh-like graphics in color and a number of powerful features. The first program, *Portfolio*, is for stock management and will be released for the IBM PC and Atari ST this fall ($249.95 for the IBM). Others in the series will be a word processor with built-in spelling checker, a spreadsheet and graphics package, and a database manager.

• Brøderbund Software of San Rafael, California announced three new programs to work with *The Print Shop*, its popular printer utility. They are *The Print Shop Graphics Library: Disk One*, *The Print Shop Graphics Library: Disk Two*, and *The Print Shop Companion*. Other new programs are *Bank Street Filer* and *Bank Street Mailer*, sequels to the *Bank Street Writer* word processor; and *Fantavision*, a special-effects generator for the Apple that uses animation technology adapted from the movie industry. ©

COMPUTE! Books
Are More Than Fun & Games

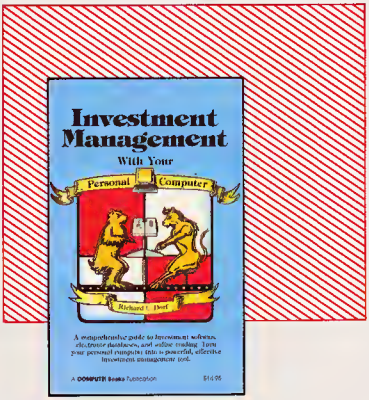
They're also challenging. Informative. Full of valuable applications for your home and business. You'll find that our newest top-quality releases contain everything you need for successful word processing, programming, or investment management.



SpeedScript: The Word Processor for the Atari

Charles Brannon, $9.95, (Disk, $12.95) ISBN 0-87455-003-3

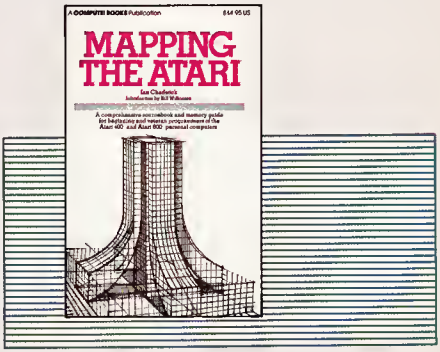
SpeedScript, the most popular program ever published by COMPUTE! Publications, is a commercial-quality word processor for the Atari 400, 800, XL, and XE computers with a minimum of 24K RAM. Included in this book are all the listings and documentation necessary to use Atari *SpeedScript*. In addition, we've included information and documentation about *SpeedScript* that have never been published before. For the price of the book, the user receives a commercial-quality word processor—an exceptional software bargain. There is a disk available for an additional $12.95 which includes programs in the book.



Investment Management with Your Personal Computer

Richard C. Dorf, $14.95, ISBN 0-87455-005-X

Covering the best available investment software for the IBM PC and Apple lines of computers, this valuable guide discusses and evaluates the programs that can help you better manage your investments. This isn't a manual of investment strategies; instead *Investment Management with Your Personal Computer* introduces all the software you'll find vital. Capsule reviews are given for programs ranging from electronic spreadsheet aids and financial databases to portfolio managers and graphics creators. The most significant software is discussed in great detail and thoroughly evaluated.



Mapping the Atari, Revised

Ian Chadwick, $16.95, ISBN 0-87455-004-1

This classic source book for users of the Atari 400 and 800 has been updated to include the 600/800XL and the new XE series Atari computers. This is the perfect supplement to the owner's manuals, and will prove valuable to both the novice and the expert. As with most COMPUTE! books, *Mapping the Atari, Revised* also contains useful appendices and indices.

For the best in home and office applications, you need COMPUTE! books.
Visit your local computer or book store for best-selling COMPUTE! books, or order directly from COMPUTE!.
Call toll-free 800-334-0868 (in NC 919-275-9809) for more information.

COMPUTE! Books are available in the U.K., Europe, Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England.

COMPUTE!™ Publications, Inc. abc
One of the ABC Publishing Companies

The Beginners Page

Tom R. Halfhill, Editor

FOR-NEXT Loops, Part 4

Over the past few columns, we've covered some of the useful techniques possible with the FOR-NEXT statement. Even if FOR-NEXT could do nothing else than what we've demonstrated, it would be indispensable. Fortunately, it can do still more—and all it takes is an optional keyword, STEP.

With STEP, you can construct "long-legged" loops and counters by varying the step size by which the counter variable is incremented. You can even make FOR-NEXT loops that count *backward*. For the sake of illustration, let's say you want to print out all the odd numbers less than 100. Without a FOR-NEXT loop, you could take this approach:

```
10 X=1
20 PRINT X
30 X=X+2
40 IF X<100 THEN 20
```

The alternative is a little shorter and easier to follow:

```
10 FOR X=1 TO 100 STEP 2
20 PRINT X
30 NEXT X
```

Without the STEP option in the FOR statement at line 10, this program would just print all the numbers from 1 to 100 because the counter variable X would be incremented by one during each pass through the loop, as usual. STEP 2 simply tells the computer to increment the counter variable X by *two* during each pass through the loop. That is, when the loop begins, X equals 1. After the first pass, X equals 3. After the second pass, X equals 5, and so on.

No Cause For Alarm

Interestingly, although the FOR statement in line 10 tells the computer to count from 1 to 100, the counter variable X actually reaches 101. You can verify this by typing PRINT X and pressing RETURN or ENTER after the program finishes. The computer reports the final value of X is 101. But don't be alarmed—the computer isn't being disobedient. Although the final STEP 2 increases X from 99 to 101, the computer still performs only 50 loops, since the upper limit specified in the FOR statement is 100 and we're stepping by twos. The program works the same if you change line 10 to FOR X=1 TO 99 STEP 2.

For even more flexibility, the STEP option lets you loop in steps of *any* increment, including fractions and negative numbers. All of the following FOR statements are valid:

```
FOR X=1 TO 1000 STEP 10
FOR X=15 TO 25 STEP 0.5
FOR X=100 TO 1 STEP -1
FOR X=1 TO 0 STEP -0.1
```

It may not be immediately apparent why you'd want to make such strange-looking loops. Mathematical operations are one typical application, but beyond that it's hard to generalize. This kind of loop is generally used to solve certain programming problems. For instance, to make a musical note decay on an Atari, you could gradually reduce the volume parameter of the SOUND statement with a backward loop (FOR X=15 TO 0 STEP −1: SOUND 0,200,10,X: NEXT X). If the note decays too quickly, you could slow it down by reducing the volume by smaller steps (STEP −0.5 or STEP −0.2) rather than embedding a second delay loop.

As you write more programs and use FOR-NEXT loops more often, eventually it will come to you in a flash that a fractional- or backward-stepping loop is exactly the solution to your problem.

BASIC Variations

FOR-NEXT statements don't follow the same rules on all computers, so you might need to consult your BASIC manual. In general, Microsoft BASICs (built into Commodore computers, the Apple, IBM, and others) let you omit the variable name from the NEXT statement if you want to. Instead of entering NEXT X you can just type NEXT. This makes the loop run faster. You can also close nested loops in Microsoft BASIC with a statement such as NEXT Z,Y,X instead of NEXT Z:NEXT Y:NEXT X or NEXT:NEXT:NEXT. (These options are not available in TI BASIC or Atari BASIC.)

Try to avoid jumping out of FOR-NEXT loops with GOTO and GOSUB statements. It's considered bad programming form, partly because it makes the program hard to follow. Besides that, a program that repeatedly exits loops with GOTO or GOSUB before the loops are finished can eventually cause some computers to crash with an out-of-memory error or the like. A few versions of BASIC have a special statement that lets you exit a loop with GOTO or GOSUB without causing any problems. In Atari BASIC the statement is POP; it's not available in Microsoft BASIC.

Almost all BASICs require the counter variable in a FOR-NEXT loop to be an ordinary numeric variable; array variables and integer variables are not allowed. An exception is IBM BASIC, which does permit integer variables. ©

The complete 1541 enhancement system . . .

STARDOS

MAKES GREAT COFFEE!!

The 1541 is the slowest disk drive on planet earth. Even simple operations seem to take forever. Quickloaders and Fastloaders that sottware-patch the operating system are vulnerable to being knocked out of memory, rendering them totally useless. Even Flashier products that require permanent moditications to the 64 and 1541 can't compete with the blinding speed of STARDOS.

STARDOS accelerates every (yes, we said every) tunction ot the 1541 disk drive. Other fast loaders only load PRG files faster. STARDOS also speeds up SEQ, REL, USR and DIRECT ACCESS files. Everything including FORMAT, VERIFY, SCRATCH, VALIDATE, INITIALIZE and COPY are much faster. In addition STARDOS adds a vast array of easy to use commands and utilities all at the touch of a key.

A sampling of STARDOS features:

- Accesses ALL types ot tiles up to 1,000% faster!
- Saves up to 300% taster than normal (with extended verify)
- DOES NOT CHANGE THE SPEED AT WHICH THE DRIVE MOTOR SPINS
- Makes your 1541 MORE RELIABLE and LESS PRONE TO BREAK DOWN or OVERHEAT!
- STARDOS is fully expandable for multiple fast disk drives
- Easy (5 minute) plug in installation. User triendly manual
- 100% compatible with sottware and serial bus peripherals
- Adds years ot lite to your disks and drive in reduced wear
- Cures a number of bugs in the Commodore 64 and 1541 disk drive including :
  - The damaging 'Head Knock' that can mis-align your 1541
  - The @: Save with replace bug!!
  - The Editor lock-up bug
- Hamess the full power of your disk drive with the built in DOS wedge
- Lock/unlock files and protect/unprotect disks trom the keyboard
- Powerful sector editor allows direct viewing of diskette
- Upgrades computer and drive to the latest Commodore specifications
- Instant access to the built in mini-wordprocessor for short notes and memos
- Built in copier copies all tile types (even relative) easily and etfectively
- Built in disk duplicator copies an entire diskettte in less than 3 minutes
- Fully expanded machine language monitor, always on line
- Quality hardware. Full 120 day repair or replace warranty
- Satisfaction guaranteed or money back—no questions asked
- (Oh, by the way, we lied, STARDOS makes ***LOUSY COFFEE***)

Simply the best. $74.95

[Take advantage ot our introduction special at $64.95]

Personalize your C64! for a $10 (non-refundable) charge we will include a 21 character power-on message with your favorite screen, border and text colors. *Example:*

M	I	K	E	'	S		C	O	M	M	O	D	O	R	E	-	6	4

Background *Black* Border *Red* Text *Green*

Write of phone for additional information.

*Trademark of Commodore Business Machines

. . . WRITE OR PHONE . . .

STARPOINT SOFTWARE

Star Route 10 Gazelle, CA 96034 [916] 435-2371

When ordering by mail:

- $64.95 + 3.00 shipping
- $64.95 + 4.00 COD orders
- Shipping out of USA $6.00
- Calif. residents add 6% sales tax
- VISA or Mastercard accepted

Please allow 4-6 weeks for delivery.

Computers and Society

David D. Thornburg, Associate Editor

Compilers, Interpreters, And Flow: Part 2

Last month I argued that one of the reasons Logo isn't used for serious applications programming is because it's not generally available as a compiler. I'm not suggesting that programmers shouldn't have interpreters—they should. However, I am suggesting that the ideal programming environment might include an interpreter for writing and testing programs, and a compiler so the completed (and mostly debugged) programs can execute much faster.

There are many fine Logo interpreters on the market, but—as this is written—only one Logo compiler: *ExperLogo* from Expertelligence (for the Apple Macintosh). To illustrate the difference between these two Logo environments, let's examine a program that uses recursion to create a fractal "sponge." (If you're interested in programming fractals, by the way, you might want to explore the subject further in two of my books, *Discovering Apple Logo* and *Beyond Turtle Graphics*, both published by Addison-Wesley.)

Here's how the program is written with an interpreter, *Apple Logo II*, running on a 128K Apple IIe or IIc:

```
to sponge:size:limit
if:size <:limit [forward:size stop]
sponge:size/3:limit
left 60
sponge:size/3:limit
left 60
sponge:size/3:limit
right 120
sponge:size/3:limit
right 120
sponge:size/3:limit
left 60
sponge:size/3:limit
left 60
sponge:size/3:limit
end
```

Once this procedure has been entered, it can be executed by entering its name with the appropriate values chosen for the variables. For example, the basic motif for the curve can be seen by entering:

```
right 90
sponge 81 81
```

To see a more detailed level of this curve, we could clear the screen and enter:

```
right 90
sponge 81 3
```

With the Logo interpreter, this picture takes 223 seconds to complete. However, as soon as the program is written it can be executed; there is no time delay before the program starts to run.

Interactive Rhythms

To write this program in *ExperLogo*, you enter the same source code into an edit window on the Macintosh screen. Then the code is selected and compiled. The compilation time for this program is 19 seconds on a 128K Macintosh (faster on a Fat Mac with 512K).

Once the program is compiled, it can be executed with the commands shown above. The compiled program draws the picture in 7 seconds—more than 30 times faster than the interpreted Logo. Of course, the compiled program executes faster partly because it's running on a 16/32-bit computer rather than on an 8-bit computer; but, as users of Macintosh BASIC will attest, interpreters can run slowly even on the Macintosh.

Is the speed gain important? For small programs, it may not be. But humans are funny creatures. We have certain rhythms in our interactions with each other and with our machines. If our technology is not operating at our pace, we become frustrated. For example, even though most photocopy machines operate pretty quickly, the perceived difference between a copier that takes 10 seconds for a copy and one that takes only 2 seconds is quite large. Those 8 seconds are just long enough to destroy the sense of flow.

In the case of computer programs we use every day, this sense of flow is even more important. When experimenting with graphic images such as those shown above, the sense of interactivity—of being able to tinker with the curves—is lost when each picture can take several minutes to create.

The Logo compiler from Expertelligence is most welcome, since it allows programmers to write commercially useful software with a powerful language—a language that in its interpreted form is often perceived as just a tool for children to draw pretty pictures.

Next month, I'll show that just as a compiler has made Logo a much more useful language, an interpreter is having the same effect on a language for which compilers are the norm—Pascal.

Thornburg welcomes letters from readers, but regrets that he cannot always provide personal answers. Correspondence should be sent in care of COMPUTE!. ©

On the Road With Fred D'Ignazio

Fred D'Ignazio, Associate Editor

Buying The Right Educational Software

What types of educational software are people buying? What kind of software do they need?

In the opinion of many educators, the most important use of the computer as a learning tool lies in improving students' thinking skills through the use of programming languages like Logo; simulations and builder kits like *The Whatsit Corporation* (Sunburst) and *Operation: Frog* (Scholastic); microworlds like *Rocky's Boots* and *Robot Odyssey I* (The Learning Company); and problem-solving software like *The Pond, SemCalc, The Factory,* and *Geometric Supposer* (Sunburst).

But the sales of these products are dwarfed by the sales of drill and practice programs and learning games. A quick glance at a recent *Billboard* chart of the ten best-selling educational packages shows that eight of them are drill and practice programs and the remaining two are learning games. Of the drill and practice programs, two teach how to type, three teach basic math skills, two help students practice for the SAT college-entry exam, and one teaches basic vocabulary and spelling skills.

A look at TESS (The Educational Software Selector), published by the Educational Products Information Exchange and the Consumers Union, shows the same dominance of drill and practice programs. Of the 7,000 programs listed in TESS, almost 70 percent are drill and practice programs, and only 8.3 percent are simulation and problem-solving programs. (For more information about TESS, write to EPIE, P.O. Box 839, Water Mill, NY 11976.)

Most experts in educational computing have been critical of drill and practice programs for years. And most experts agree that problem-solving and simulation software is the most challenging and interesting software for anyone learning on a computer. If this is true, why are companies producing so much drill and practice software? And, more importantly, why do people prefer it?

Wary Adults

The answer is that most parents (and many teachers) are not ready for new kinds of software that teach new skills in new, unfamiliar ways. They don't understand how the programs work or what they're supposed to teach, or why it's important, and they don't see where the programs fit into their children's learning. And since they don't see a need for the programs, they don't buy them.

This is a natural reaction. For most people, computers are still a strange, almost alien, new medium. Many parents are still uncomfortable having a computer in their home. And many teachers, too, feel privately fearful of computers. They see the computer as a threat—a means to automate them out of a job. The more the computer's role in the classroom grows, the more they see their own role being eroded.

In addition, problem-solving and thinking-skill software is an unfamiliar, new application of computers. We have a new medium (computers) trying to teach new concepts (logic and thinking skills) using new methods (microworlds, simulations, etc.). This is too much novelty for the average consumer—whether that consumer is a parent or a school system. As a result, most consumers are buying drill and practice programs and learning games because at least this way they see the computer teaching practical, necessary, and familiar skills—using a nonthreatening, understandable approach. And in the classroom, since the skills are familiar, the programs that teach them are more easily integrated into a teacher's lesson plan and curriculum. A program that teaches a child some spelling words can slip effortlessly into a curriculum, but what does a teacher do with a program that teaches a child how to think?

For the present, most parents will be buying and using drill and practice software and learning games, and ignoring problem-solving and simulation software. Does this mean that companies should stop producing these more challenging, yet less successful programs? Hardly. Instead, educators and software companies need to launch a major effort to communicate to parents and teachers the importance of the new kinds of software. To do this, software companies must demonstrate to parents and teachers why learning these skills is important, and how the software fits into their children's learning curriculum. ©

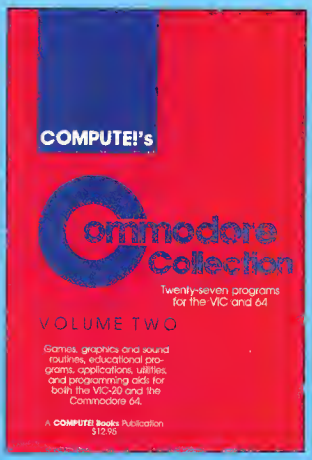
For Commodore owners, the latest collections from COMPUTE! Books

COMPUTE!'s Commodore Collection, Volume 1
Twenty-eight original programs for the VIC and 64

Edited

The *Commodore Collection*, an anthology of 28 previously unpublished programs for the VIC-20 and Commodore 64, continues the COMPUTE! tradition of practical programs in easy-to-use form. Includes exciting games, stimulating educational programs, valuable utilities, and useful applications to make your Commodore computer an exciting part of every day.

$12.95 ISBN 0-942386-55-8



COMPUTE!'s Commodore Collection, Volume 2
Programs for the VIC and 64

Edited

Exciting games, sophisticated applications, versatile educational routines, and helpful programming aids for the VIC-20 and Commodore 64 highlight this second volume in COMPUTE!'s Commodore Collection series. Included are some of the best articles and programs from recent issues of *COMPUTE!* and *COMPUTE!'s Gazette*, as well as many programs published here for the first time. Designed for Commodore computer users of all levels, it's a book that every VIC or 64 owner will want to have.

$12.95 ISBN 0-942386-70-1

To charge your order, call toll free 800-334-0868 or send in this coupon with your payment to COMPUTE! Books, P.O. Box 5058, Greensboro, NC 27403.

Please add $2.00 shipping and handling per copy ordered.

☐ Payment enclosed (check or money order)
☐ Charge: ☐ Visa ☐ MasterCard ☐ American Express

Account No. _____ Exp. Date _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

Please allow 4–6 weeks for delivery

7581101

___copies of *COMPUTE!'s Commodore Collection, Volume 1* @ $12.95 ea. _____

___copies of *COMPUTE!'s Commodore Collection, Volume 2* @ $12.95 ea. _____

Shipping & handling, $2.00/book _____

Total payment _____

COMPUTER MAIL ORDER

THE CMO ADVANTAGE

- THE BEST PRICES!
- Next day shipping on all in stock items.
- Free easy access order inquiry.
- Orders from outside Pennsylvania and Nevada save state sales tax.
- Free technicial support with our factory trained technical staff.
- There is no limit and no deposit on C.O.D. orders.
- There's no extra charge for using your credit card. Your card is not charged until we ship.
- No waiting period for cashiers checks.
- We accept purchase orders from qualified corporations. Subject to approval.
- Educational discounts available to qualified institutions.
- FREE CATALOG MEMBERSHIP.

ORDER LINE

1-800-233-8950
In PA 1-800-242-4215
CUSTOMER SERVICE & TECH SUPPORT
1-717-327-1450
Dept. A208

MAILING ADDRESS

EAST
Dept. A208, 477 E. Third St.
Williamsport, PA 17701

WEST
Dept. A208, P.O. Box 6689
Stateline, NV 89449

MEMBER DIRECT MARKETING ASSOCIATION

CREDIT CARDS

VISA MasterCard American Express Diners Club

SHIPPING

Add 3%, minimum $5.00 shipping and handling on all orders. Larger shipments may require additional charges.

All items subject to availability and price change.

Returned shipments may be subject to a restocking fee.

CANADIAN ORDERS

1-800-268-3974
Ontario/Quebec

1-800-268-4559
Other Provinces

1-416-828-0866
In Toronto

TELEX: 06-218960

2505 Dunwin Drive,
Mississauga, Ontario
Canada L5L1T1

All prices shown are for U.S.A. orders.
Call The Canadian Office for Canadian prices.

HOME COMPUTERS

APPLE

APPLE IIe	CALL
APPLE IIc	CALL
MacINTOSH	CALL
IIc LCD Display	CALL

Macintosh Software

Lotus Jazz	CALL
Microsoft Excel	CALL
Microsoft Business Pak	$375.00
Living Videotext ThinkTank 512	$159.00
Manhatten Ready, Set, Go	$79.99
Craighton Development Mac Spell	$69.99
Monogram Dollars & Sense	$99.99
Peachtree Back to Basics - GL	$109.00
PFS File & Report (New Version)	$129.00
Silicon Beach Airborn	$25.99

ATARI

130XE (128K)	**CALL**
520ST (512K)	**CALL**

ATARI 600XL CLOSEOUT $49.99
WHILE SUPPLIES LAST

800XL 64K	CALL
850 Interface	$119.00
1010 Recorder	$49.99
1020 Color Printer	$79.99
1025 Dot Matrix Printer	$199.99
1027 Letter Quality Printer	$269.99
1030 Direct Connect Modem	$59.99
1050 Disk Drive	$179.99
Touch Tablet	$64.99
7097 Atari Logo	$69.99
4018 Pilot (Home)	$57.99
5049 VisiCalc	$49.99
4011 Star Raiders	$12.99
4022 PacMan	$16.99
8036 Atari Writer	$79.99

BOARDS FOR ATARI

Axlon 32K	$39.99
Axlon 48K (400)	$69.99
Axlon 128K	$269.99
Microbits 64K (600)	$109.00
Bit 3 Full View 80	$229.00

commodore

NEW Commodore 128, LCD	CALL
SX-64 Portable	CALL
Commodore Plus 4	$199.00
CBM 64	$149.00
C1541 Disk Drive	$199.00
C1530 Datasette	$39.99
M-801 Dot Matrix Printer	$189.00
M-802 Dot Matrix/Serial	$219.00
MCS 803 Dot Matrix	$179.00
C1802 Color Monitor	$199.00
C1660 Auto Modem	$59.99
DPS 1101 Daisy Printer	$339.00

Professional Software

Fleet System II w/Spell	$49.99
Trivia Fever	$29.99
Word Pro 4 Plus/5 Plus each	$239.00
Info Pro	$179.00
Administrator	$399.00
Power	$69.99

pfs:

File (64)	$59.99
Report (64)	$49.99

Precision Software

Superbase 64	$54.99

BATTERIES INCLUDED

PaperClip w/Spell Pack	$79.99
The Consultant DBMS	$69.99
Bus Card II	$139.00
80 Col Display	$139.00

PORTABLE COMPUTERS

HEWLETT PACKARD

41CV	$169.99
41CX	$249.99
HP 71B	$419.99
HP 11C	$62.99
HP 12C/15C/16C	$89.99
HP 75D	$999.99
HPIL Module	$98.99
HPIL Cassette or Printer	$359.99
Card Reader	$143.99
Extended Function Module	$63.99
Time Module	$63.99

We stock the full line of HP calculator products

NEC

PC-8401	CALL
PC-8201 Portable Computer	$289.00
PC-8231 Disk Drive	$599.00
PC-8221A Thermal Printers	$149.00
PC-8281A Data Recorder	$99.99
PC-8201-06 8K RAM Chips	$105.00

SHARP

PC-1350	$159.99
PC-1261	$159.99
PC-1260	$109.99
PC-1500A	$165.99
PC-1250A	$88.99
CE-125 Printer/Cassette	$128.99
CE-150 Color Printer Cassette	$171.99
CE-161 16K RAM	$134.99

DRIVES

HARD

ALLOY

PC Stor	CALL

IDEAssociates

5 meg Removable/Internal	$1399.00
10 meg Fixed/Internal	$1249.00
15 meg 5 Removable/10 Fixed	$2149.00
25 meg 5 Removable/20 Fixed	$2499.00

i-MEGA

10 Meg Bernoulli Box	$2099.00
20 meg Bernoulli Box	$2599.00
5 meg "MacNoulli"	$1599.00

TALLGRASS TECHNOLOGIES

COMMITTED TO MEMORY

12, 25, 35, 50, 80 meg (PC) from $1499.00

FLOPPY

INDUS

Apple GT	$209.00
Atari GT	$239.00
C-64 GT	$259.00

µ-SCI

A1.5 Apple	$199.00
A2 Apple	$199.00

VISI

SD1 C-64 Single	$219.00
SD2 C-64 Dual	$469.00

tandon

320K 5¼" (PC)	$119.00

Qume

320K 5¼"	$99.99

ACCESSORIES

KEYTRONICS

KB5150, 5151, KB5152B, KB5153	CALL
5151 Jr., 5149 Jr.	CALL

MEMORY CHIPS

4164 RAM Chips	(ea.) $1.99

Microphonics

Pronounce Speech Input System	$399.00

MODEMS

ANCHOR

Volksmodem	$59.99
Volksmodem XII	$189.99
Mark II, Serial	$79.99
Mark VII (Auto Ans/Auto Dial)	$99.99
Mark XII (1200 Baud)	$259.00

Hayes

Smartmodem 300	$145.00
Smartmodem 1200	$389.00
Smartmodem 1200B	$359.00
Smartmodem 2400	$699.00
Micromodem IIe	$249.00
Smart Com II	$89.99
Chronograph	$199.00
Transet 1000	CALL

AST

Reach 1200 Baud Half Card	$399.00

mee MICROBITS

MPP-1000E AD/AA (Atari)	$79.99
MPP-1064 AD/AA (C-64)	$69.99

Novation

Smart Cat Plus	$319.00
J-Cat	$99.99
Smart Cat 103/212	$369.00
Novation 2400	CALL
212 AutoCat II	$499.00
Apple Cat II	$229.00
212 Apple Cat II	$379.00
Apple Cat 212 Upgrade	$229.00
Macmodem	$319.00

QUADRAM

Quadmodem II 300/1200	$339.00
300/1200/2400	$499.00

TELELEARNINO

C64 300 Baud (Closeout)	$39.00

ZENITH

ZT-1	$339.00
ZT-10	$309.00
ZT-11	$369.00
Z-22 Video Data Terminal	$529.00

DISKETTES

maxell.

3½" SS/DD	$39.99
3½" DS/DD	$54.99
5¼" MD-1 w/Hardcase	$17.99
5¼" MD-2 w/Hardcase	$23.99
5¼" MD-2-HD for AT	$44.99

Verbatim.

5¼" SS/DD	$21.99
5¼" DS/DD	$29.99
Disk Analyzer	$24.99

Dennison

Elephant 5¼" SS/SD	$13.99
Elephant 5¼" SS/DD	$15.99
Elephant 5¼" DS/DD	$16.99
Elephant EMSP 5¼"	$24.99

Bib

5¼" Disk Head Cleaner	$14.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Flip-in-File 10	$3.99
Flip-in-File 50	$17.99
Flip-in-File 50 w/lock	$24.99
Flip-in-File (400/800 ROM)	$11.99

AMARAY

50 Disk Tub 5¼"	$9.99
30 Disk Tub 3½"	$8.99

GRAPHICS

Koala

IBM	$89.99
Apple/Franklin	$79.99

Polaroid

Palette	$1299.00

CALL TOLL-FREE

MONITORS

AMDEK

300 Green.....$129.00
300 Amber.....$139.00
310 Amber IBM-Plug.....$169.00
Color 300 Composite.....$239.00
Color 500 Composite/RGB.....$399.00
Color 600 Hi-Res (640x240).....$399.00
Color 700 Hi-Res (720x240).....$499.00
Color 710 Long Phosphor.....$579.00

NAP

12" Amber/Green Composite.....$99.99
12" Amber/Green TTL.....(ea.) $119.00

NEC

JB 1260 Green.....$59.99
JB 1201/1205 (ea.).....$99.99
JB 1270 Green.....$139.00
JB 1275 Amber.....$149.00
JB 1280 G TTL/1285 A TTL.....$149.00
JC 1460 Color.....$269.00
JC 1410 RGB.....$669.00

PRINCETON

MAX-12E Amber.....$189.00
HX-9 9" RGB.....$469.00
HX-9E Enhanced.....$519.00
HX-12 12" RGB.....$469.00
HX-12E Enhanced.....$559.00
SR-12 Hi-Res.....$599.00
SR-12 P Enhanced.....$649.00

TAXAN

115 12" Green Mono.....$119.00
116 12" Amber Mono.....$119.00
121 Green TTL.....$139.00
122 Amber TTL.....$149.00
400 Med-Res RGB.....$299.00
420 Hi-Res RGB (IBM).....$429.00
440 Ultra Hi-Res RGB.....$589.00

QUADRAM

8400 Quadchrome I.....$479.00
8410 Quachrome II.....$429.00
8420 Amberchrome.....$179.00
8500 Quad Screen.....$1499.00

ZENITH

ZVM 122/123.....$79.99
ZVM 124 IBM Amber.....$149.00
ZVM 130 Color.....$269.00
ZVM 131 Color.....$279.00
ZVM 133 RGB.....$429.00
ZVM 135 RGB/Color.....$459.00
ZVM 136 RGB/Color.....$599.00
1220, 1230, 1240.....CALL

INTERFACES

PRACTICAL PERIPHERALS

Graphcard.....$79.99
Senall Card.....$99.99
Microbuffer II+.....$169.00
Microbuffer 32K.....$189.00

QUADRAM

Microfazer.....from $139.00
Efazer (Epson).....from $79.99

Orange Micro

Grappler CD (C64).....$99.99
Grappler + (Apple).....$89.99
Grappler 16K+ (Apple).....$159.00

DIGITAL DEVICES

Ape Face (Atari).....$49.99
U-Print A (Atari).....$54.99
U-A16/Buffer (Atari).....$74.99
U-Call Interface (Atari).....$39.99
U-Print C (C64).....$49.99
P-16 Print Buffer.....$74.99

MPP MICROBITS

MB1150 Parallel (Atari).....$79.99
MPP-1150 Parallel (Atari).....$69.99
MP-1150XL (Atari 1200XL).....$69.99
MicroStuffer 64K Print Buffer...$109.00

PRINTERS

OKIDATA

AT-100 Atari Interface Printer...$139.00
AT-550 Atari Dual Mode.....$249.00
GP-100 Parallel Interface.....$189.00
GP-700 Color Printer.....$449.00
GP-550 Parallel Printer.....$239.00
Elite 5 Letter Quality.....$239.00
Elite 5 C64 Interface.....$249.00
Penman 3-pen.....$289.00

CITIZEN

MSP-10 (80 col.).....$349.00
MSP-15 (132 col.).....$499.00
MSP-20 (80 col.).....$489.00
MSP-25 (132 col.).....$679.00

C.ITOH

Prowriter 7500.....$219.00
Prowriter 8510P.....$299.00
Prowriter 8510 NLQ.....$329.00
Prowriter 1550P.....$469.00
F10-40P Starwriter.....$869.00
F10-55 Printmaster.....$1049.00
Prowriter 8510-NLQ.....$329.00

COMWRITER

ComWriter!! Letter Quality.....$399.00

corona

Lazer LP-300.....$2799.00

DIABLO

D25 Daisywheel.....$599.00
630-109 Daisywheel.....$1749.00
D801F Daisywheel.....CALL

daisywriter

2000.....$749.00

EPSON

RX-80, FX-80+, LX-80, JX-80.....CALL
FX-100+, RX-100, LQ1500.....CALL
Homewriter 10.....CALL
NEW! LX-90, SQ-2000, DX10, DX20, HS-80.....CALL

JUKI

6000 Letter Quality.....CALL
6100 Letter Quality.....CALL
6300 Letter Quality.....CALL

NEC

8027 Transportable.....$299.00
2000 Series.....$699.00
3000 Series.....$1099.00
8000 Series.....$1499.00
ELF 360.....$449.00

OKIDATA

84, 182, 192, 193, 2410.....CALL
Qkimate 10 (Specify C64/Atari).$199.00
Qkimate 20 (IBM).....CALL

OLYMPIA

Needlepoint Dot Matrix.....$299.00
Compact RO.....$339.00
Compact 2.....$369.00

Panasonic

KX1090.....$199.00
KX1091.....$279.00
KX1092.....$409.00
KX1093.....$599.00

QUADRAM

Quadjet.....$399.00

SILVER-REED

500 Letter Quality.....$279.00
550 Letter Quality.....$419.00
770 Letter Quality.....$759.00

star

SB/SD/SG/SR Series.....CALL
Powertype Letter Quality.....CALL

TOSHIBA

1340 (80 column).....$599.00
P351 (132 column).....$1299.00

PC COMPATIBLES

IBM PC SYSTEMS

Configured to your specification. Call for Best Price!

IBM-PC, IBM-PC II, IBM-XT, IBM-AT

COLUMBIA DATA PRODUCTS

2220 Dual Portable.....$1499.00
4220 Dual Desktop.....$1399.00
4620 10 meg Desktop.....$2299.00

AT&T

Safari (7300).....CALL
6300.....CALL

SOFTWARE FOR IBM

ALPHA

Electronic Desk.....$199.00

ASHTON-TATE

Framework.....$399.00
dBase II.....$299.00
dBase III.....$389.00

BORLAND INTERNATIONAL

Turbo Pascal 3.0.....$49.99
Sidekick.....$39.99

ELECTRONIC ARTS

Get Organized.....$69.99
Cut -n- Paste.....$39.99
Music Construction.....$29.99
One -on- One.....$29.99
Financial Cookbook.....$34.99

Harvard Software Inc.

Harvard Project Manager.....$209.00
Total Project Manager.....$269.00

Human Edge™

Communication Edge.....$99.99
Management Edge.....$119.00
Negotiation Edge.....$139.00
Sales Edge.....$119.00

IMSI

PC Paintbrush.....$94.99

Lotus

Symphony.....$439.00
1-2-3.....$309.00

MicroPro

WordStar 2000.....$249.00
WordStar 2000+.....$319.00

MICROSOFT.

Flight Simulator.....$39.99
MultiPlan.....$129.00

MICROSTUF

Crosstalk.....$89.99

MICRORIM

R:Base 4000.....$249.00
R:Base 5000.....$399.00
Clout 20.....$129.00

MultiMate

Multi Mate.....$249.00

Peachtree Software

PeachPack (GL/AP/AR).....$199.00

pfs:

IBM/APPLE

Access (NEW).....$79.99
Write/Graph/File/Plan.....(ea).$79.99
Report.....$74.99
Proof.....$59.99
Mac Software.....CALL

Professional Software

PC Plus/The Boss.....$249.00

SOFTWARE GROUP

Enable.....$369.00

SORIUM/IUS

Accounting AP/AR/GL/INV/QE.....(ea).$295.00
Supercalc III.....$195.00
EasyWriter II System.....$195.00
Super Project.....$195.00

SPI

Open Access.....$379.00

SSI Software

Word Perfect.....$239.00

synapse

File Manager (IBM).....$39.99

THOUGHTWARE

Trigger.....$289.00
Sell, Sell, Sell Training.....$299.00
Application.....$179.00

ZENITH

PC-151-21 Single Desktop.....CALL
PC-151-52 Dual Desktop.....CALL
PC-151-53 10 meg Desktop.....CALL
PC-161-21 Single Portable.....CALL
PC-161-52 Dual Portable.....CALL
Z-200 (AT).....CALL
171 (Portable).....CALL
138 (Transportable).....CALL
148 (DeskTop).....(NEW).....CALL

corona

PPC400 Dual Portable.....$1799.00
PPCXT 10 meg Portable.....$2399.00
PC40022 10 meg Desktop.....$1999.00

SANYO

MBC 550-2 Single Drive.....$699.00
MBC 555-2 Dual Drive.....$969.00
MBC 775 Portable.....$1799.00
MBC 511 10 meg.....CALL
MBC 675 Portable.....CALL
NBC 880 Desktop.....CALL

MULTIFUNCTION CARDS

AST

Six Pack Plus.....$239.00
Mega Plus II.....$269.00
I/Q Plus II.....$139.00
Advantage-AT.....$399.00
Graph Pak.....$599.00
Monograph Plus.....$399.00
Preview Mono.....$299.00
PC Net Cards.....$379.00
5251/11 On-line.....$799.00
5251/12 Remote.....$579.00
3780 Emulation Card.....$639.00
BSC Bisync.....$489.00

dca

IRMA 3270.....$879.00
IRMA Print.....$999.00

EAGLE

Mono Card.....(Closeout).....$99.99

EVEREX

Color Card (Graphics Edge).....$299.00
Magic Card.....$199.00

HERCULES

Graphics.....$299.00
Color.....$159.00

IDEAssociates

IDEAmax - ZPR, 64K, C, S, P.$229.00
IDEAmini - YPR, C, S, P.....$189.00
IDEAminimax - MPR 128K.....$229.00
IDEAshare Software.....$219.00
IDEA 5251.....$699.00

MYLEX

The Chairman.....$489.00

PARADISE

Modular Graphics Card.....$274.00
Multi Display Card.....$289.00
Five Pack C, S.....$159.00

PLANTRONICS

Color Plus.....$369.00

TECMAR

Captain - 64.....$239.00
Captain Jr. 128K.....$339.00
Graphics Master.....$469.00

QUADRAM

Quadport-AT.....$119.00
Quadmeg - AT (128K).....$349.00
The Gold Quadboard.....$449.00
The Silver Quadboard.....$239.00
Expanded Quadboard.....$219.00
Quad 512+.....$229.00
Liberty.....$309.00
QuadSprint.....$499.00
QuadLink.....$399.00
Quadcolor 1.....$199.00
QuadJr. Expansion Chassis.....$489.00
Expansion Chassis Memory.....$199.00
Quadmem. Jr.....$199.00
Chronagraph.....$79.99
Parallel Interface Board.....$64.99

COMPUTER MAIL ORDER

1-800-233-8950

HOTWARE: Software Best Sellers

This Month	Last Month	Title	Publisher	Remarks	Apple	Atari	Commodore	IBM	Macintosh
Entertainment									
1.	3.	*Flight Simulator II*	SubLogic	Aircraft simulotion	•	•	•		
2.	2.	*The Hitchhiker's Guide To The Galaxy*	Infocam	Comic adventure strotegy gome	•	•	•	•	•
3.	1.	*Flight Simulator*	Micrasaft	Aircroft simulotion				•	
4.	4.	*F-15 Strike Eagle*	MicraProse	Air combat simulotion	•	•	•		
5.		*Summer Games II*	Epyx	Action sparts game	•	•	•	•	
Education									
1.	3.	*New Improved MasterType*	Scorboraugh	Typing instructian pragrom	•	•	•	•	•
2.	1.	*Typing Tutor III*	Simon & Schuster	Typing instruction program	•		•	•	•
3.	2.	*Math Blaster!*	Davidson	Introductary math progrom, ages 6-12	•	•	•	•	
4.	5.	*Early Games*	Springbaord	Educatianol gomes, ages 2-6	•	•	•	•	
5.	4.	*Word Attack!*	Davidsan	Ward meoning ond usage pragrom, grades 4-12	•	•	•	•	
Home Management									
1.	1.	*Print Shop*	Brøderbund	Do-it-yaurself print shap	•	•	•		
2.	2.	*Print Shap Graphics Library*	Brøderbund	100 additional graphics	•	•	•		
3.	5.	*Bank Street Writer*	Brøderbund	Ward Processar	•	•	•	•	
4.		*HamePak*	Botteries Included	Word processing, telecammunicotsions, & doto monogement	•	•	•	•	
5.	3.	*Andrew Tobias's Managing Your Maney*	Meca	Home finonciol program				•	

Telecomputing Today

Arlan R. Levitan

SIGs: Behind The Scenes

It appears that 1985 is rapidly becoming a banner year for electronic Special Interest Groups (SIGs). Unfortunately, rapid growth is not without its problems.

Electronic SIGs (also known as discussion forums) are like computerized clubs where people with a common interest get together online to exchange information, ideas, comments, complaints, and public domain computer programs. Usually they are operated by commercial information services such as CompuServe. Besides computer-oriented SIGs, there are groups for doctors, lawyers, educators, and other professionals, not to mention enthusiasts of travel, cooking, literature, rock 'n' roll, politics, human sexuality, skiing, and ham radio. There are SIGs for every taste, and new ones are added every day.

SIGs are usually divided into three areas. The most active area is a message system subdivided into several sections by topic where users can read and leave messages. The oldest messages are overwritten when a new message is added, so each message tends to have a life of a week or less.

Each topic section also has a file access database where important messages may be stored more permanently as text files, and where users leave free copies of public domain software.

Finally, each SIG also has a conference area so online users can exchange messages in an immediate mode. Everything you type appears on the screens of others participating in the conference.

Online Bucks

SIGs generate extra income for information services because they encourage more online usage. But how valuable are SIGs to an information service's bottom line? Frankly, most SIGs generate only a modest amount of revenue, and some don't break even. There are, however, a significant number of SIGs that are monstrous hits, largely due to the work of their sysops (system operators). In fact, each of the top SIGs generates well over a million dollars a year! That can mean big bucks for the head sysop. It's not uncommon to find the sysop (or company sponsoring the SIG) pulling down at least 5 percent of the gross—$50,000 a year or more. The assistant sysops generally work for "fame and glory" and get free time on the SIG for their efforts.

Although SIGs used to be the sole province of CompuServe, the revenues that they've been pulling in lately have prompted Delphi, The Source, and other services to get into the act. And the competition is just beginning to heat up. The lure of substantial amounts of SIG-generated money can do funny things to people.

For example, the entire staff of the Commodore SIG on one service recently jumped ship and set up shop with a competitor. The defection wasn't a secret—the sysops used the SIG message base to let members know where they had gone. And within the last year, the head sysop of another popular SIG had to contend with a palace coup staged by his assistant sysops. After the smoke cleared, there were two SIGs instead of one—a SIG headed by the original sysop and another by the rebels. Keeping the SIG peace might soon be a job for the United Nations.

Fortunately, most SIGs are very well run and perform a valuable function, acting as information clearinghouses. But as useful as they may be, there's no point in paying through the nose to enjoy the benefits of regular involvement in a SIG. Here are some general tips that can help you save money when participating in a SIG.

Be Selective

Don't try to read every new item in the message area. Many of the more popular SIGs can turn over a thousand messages or more in three or four days. Pick one or two sections with topics that appeal to you and stick with them.

Many SIGs let you read streams of messages without pausing for a response from you between items. To take full advantage of this feature, download the messages you want and read them offline. If you wish to reply to a message or leave a new one of your own, write your text offline as well; you can send the entire message in a matter of seconds rather than pecking away online. (See last month's column for hints on this technique.)

Unless you have a burning question that can't be answered by the folks who access the message section, pass up regularly attending special online conferences. Complete transcripts of the conferences are usually available in the file access areas shortly afterward. Given the speed that most people type, the text of a two-hour conference can usually be downloaded in about five minutes. If you can't resist conferences, don't bother to sign on at 1200 or 2400 bps if the service charges hourly premiums for these higher speeds. Ol' 300 baud is just fine for conferences and will keep the hourly rate you pay at a minimum.

Finally, don't ignore using the public domain program library of your local computer user group as an alternative to downloading files from SIGs. An entire diskful of programs from a user group usually costs no more than an hour of time on a SIG. ©

INSIGHT: Atari

Bill Wilkinson

Atari Input/Output

Much of what I'm about to discuss this month has appeared in this column before. And the bulk of this information can also be found in the *Atari Technical Reference Manual*—presuming you can read "techlish." But this intro is necessary so we can start talking about the meat of our subject next month.

Still with me? Let's go. Atari's operating system (OS)—which, like the OS in any eight-bit computer, takes up the bulk of Read Only Memory (ROM)—is really a thing of beauty. In fact, it may be the only *consistent* OS to be found in any microcomputer, short of those sporting UNIX or its derivatives. CP/M and MS-DOS are such kludges that most commercial programs bypass the OS. (That's why there are so many "almost PC-compatibles.") The Commodore 64's operating system comes close, but its disk input/output is difficult at best. And Apple's ProDOS manual states that "users desiring to perform I/O to devices other than the disk drive are on their own!"

Atari users, on the other hand, enjoy a system with such complete support that, for most programs, all necessary input/output operations can be executed by calling a single subroutine! That subroutine is called, appropriately, *Central Input/Output* (CIO). By calling CIO with the proper values in certain memory locations and the proper pointer in the 6502's X register, your programs can perform such diverse operations as formatting a disk, drawing a line on the graphics screen, fetching a keystroke from the keyboard, sending output to the printer, or reading 25,000 bytes from a disk file.

Yet, CIO is invisible to most Atari users. For example, many of the capabilities which magazine and newsletter articles attribute to BASIC are not part of BASIC at all. None of the graphics (including the so-called BASIC graphics modes) in Atari BASIC are actually performed by BASIC. Instead, BASIC simply translates the graphics command into a call to CIO. Atari developed this system in 1978, and it wasn't until the Macintosh appeared that such a revolutionary concept was repeated in a popular computer.

Generally, you have to become a machine language programmer to appreciate and use all the features of CIO. So why read any of this, then? Because calls to CIO can't perform *every* input/output task possible on Atari computers. What can't CIO do? First, let's take a glance at what it can do.

Calling CIO

When CIO is called by a program, it expects the X register to contain a pointer to an *Input/Output Control Block* (IOCB). IOCBs are blocks of memory 16 bytes long which control CIO functions. The pointer value for the X register is easily calculated—it's actually the BASIC file number (as in OPEN #1,4,0,"K:") multiplied times 16, because there are 16 bytes per IOCB. One of the bytes within the IOCB then tells CIO what function the program is requesting.

There are seven fundamental functions available: OPEN, CLOSE, STATUS, PRINT, INPUT, Block PUT, and Block GET. In addition, there are some *extended functions*. BASIC programmers are familiar with these because of the XIO statement, which allows you to call the functions from BASIC. But several other BASIC statements (including NOTE, POINT, DRAWTO, and LOCATE) access the CIO extended functions, too.

After CIO examines the IOCB and determines which function is being requested, it decides which device (keyboard, disk, screen, etc.) should service the request. Then it calls an appropriate routine within the *device driver* for that device. (For example, the Disk Operating System—or more properly, the File Management System—is the device driver for the disk drive.) If the request is for an extended function, it is passed on unchanged to the device driver.

Well, with 256 possible command values, you would think that there isn't any request, however bizarre, which couldn't be serviced via CIO. In theory, true. In reality, you have to stop adding functions somewhere or you run out of memory. Thus Atari's CIO-based graphics have no function for drawing a circle, and DOS provides no command to format a disk without also writing a boot and directory.

If you want to draw a circle, you can write a routine to calculate and PLOT points or change screen memory directly. If you want to mess with the disk drive, though, you have to learn about another routine within the Atari ROMs, *Serial Input/Output*.

The Mysterious SIO

SIO—which lets Atari computers talk to devices (such as printers and disk drives) which hook up to the serial bus—has acquired an undeserved aura of mystery. Actually, though, in some ways it is easier to call SIO than it is to call CIO!

For example, there is only one SIO "device" and only one Device Control Block (DCB). So even the X register pointer required by CIO isn't necessary when calling SIO. Intrigued? I hope so, because it's time to sign off for now. But after this brief overview, we're ready for next month's column: We'll show how to write a program to call SIO. ©

IBM Personal Computing

Donald B. Trivette

New Life For Aging PCs

This month's COMPUTE! is full of new wares from the Summer Consumer Electronics Show. But most of us can't afford to toss out the old and bring in the new—at least not yearly—so let's look at an interesting alternative.

One way to give new life to an aging PC is with a PCturbo board by Orchid Technology (47790 Westinghouse Drive, Fremont, CA 94539). PCturbo, which installs in one of the PC's expansion slots, contains a whole new computer *and* a whole new memory. It's like having a new computer inside an old computer. The old computer is the original IBM PC with its memory and Intel 8088 microprocessor. The new computer is completely contained on the PCturbo board; it has an Intel 80186 microprocessor with up to 640K RAM.

You don't lose your old PC in the bargain—a PC with PCturbo is like two separate computers. In PC mode, the computer runs just as it always did. But in PCturbo mode, the machine zips through most programs three to four times faster. Two new DOS-like commands—TURBO and GOPC—switch from one mode to another.

The PCturbo board executes all the usual program instructions, but handles input/output in a special way. For example, when a disk operation generates an input/output interrupt, the PCturbo's 80186 processor passes the I/O task along to the slower 8088 processor. That means the 80186 can run a program at full speed without stopping to process time-consuming interruptions from the keyboard, screen, disk, and printer.

Nor is the PC's original memory left idle. Software supplied with PCturbo allows that memory to be used as one or more electronic RAM disks and as a *disk cache* (pronounced like "cash"). Disk caching has been around for years as a way to speed up I/O on mainframe computers. No matter how fast disk drives are, memory is three to four times faster. The PCturbo disk-caching software automatically moves blocks of data (called sectors) from the disk to the cache (the old PC memory) so the data will be there when needed. In fact, the caching software anticipates requests: It moves the *next* disk sector into memory as well. All this is completely transparent to the application software—it still thinks it's reading and writing data on a floppy disk.

A Dramatic Speed-Up

How well does disk-caching work? To find out, I plugged a PCturbo board into my three-year-old 320K PC and invited over a friend.

Henry and his wife own an interior design firm; they use their PC with *Lotus 1-2-3* to keep track of chairs, tables, wallpaper, fabrics, taxes, and all the other charges for a job. I had heard Henry complaining about the spreadsheet for a modest residence he is doing for a surgeon. The spreadsheet is 28 columns wide by 389 rows deep—about 170,000 opulent bytes. When something is added—say a Ming vase—it takes the program 15.5 seconds to recalculate. (We all should have to spend a half-million dollars at four items per minute!) With the computer in PCturbo mode, the recalculations were cut to six seconds. That's a speed increase of 250 percent (or six *more* items per minute).

My own problems more often involve checking the spelling in a manuscript rather than spending money more rapidly. I recalled a 2,049-word article (it took the PC three seconds to count the number of words; PCturbo, one second) and set *Word Proof*, the IBM spelling checker, to work. In 41 seconds, PCturbo verified that the article had no errors; in regular PC mode, the job took 107 seconds. To speed things up even more, I moved *Word Proof*'s 125,000-word dictionary from a floppy disk to an electronic RAM disk created in the regular PC memory. PCturbo polished off all 2,049 words in 18 seconds (almost 114 words per second); the PC by itself took 77 seconds—more than four times longer.

And how much does it cost to find spelling errors faster than Superman? The invoice from Orchid Technology is $747 for the PCturbo board with 256K of memory, and $375 for the optional 384K memory module to reach a total 640K. The fully loaded board—PCturbo with 640K—costs $1,122. Not bad for an accessory that almost turns a three-year-old PC into a PC AT.

Almost, that is, because the PC won't do a few things in PCturbo mode. IBM BASIC is an IBM proprietary product built into a chip on the PC's main circuit board; it's not available to PCturbo, so you must switch to PC mode to run BASIC or BASICA. But compiled BASIC runs just fine with PCturbo. Also, the PCturbo board I tested wouldn't run communications programs (it couldn't find the modem). However, Orchid says it has a solution to that in the works. Somehow it's comforting to know there's lots of life left in the old PC.

Odds And Ends

The second edition of IBM's *Directory of Personally Developed Software* has been out for several months. To get a free copy, call 1-800-IBM-PCSW (a.k.a. 1-800-426-7279). The new directory lists 58 software products ranging in price from $15 to $150, although most cost about $20. There are some real bargains here. ©

Programming the TI

C. Regena

Trivia Quiz

There has been quite a big fuss about the trivia-type question and answer games—Trivial Pursuit and the various takeoffs. I'm certain that you've also seen ads for computer versions of these board games. Have you ever wanted to buy one? Wait! Write your own. This month's program, "Trivia Quiz," gives you the basic structure for a question and answer game.

The main part of either the board games or the computer versions is the bank of questions. Of course, we can't publish a program with questions already included because it would take up the whole magazine. To make your own game, you must invent your own questions. Teachers can use this program for a basic essay-type quiz on any subject, and families can think up questions that appeal to their interests.

The computer is used to shuffle the questions, or to randomly choose one question at a time and give its corresponding answer. Once a question comes up, it is not used again.

I've previously published programs for a matching quiz and a multiple-choice quiz. This program creates a fill-in-the-answer or essay quiz. You can even print out copies of tests that have the questions in a different order for each person.

Modifying The Program

In each of the program's DATA statements, first write a question, then type a comma, then put the answer. Remember to adjust the spacing so it looks nice on the screen (no words split across screen boundaries). If you need commas within the question or the answer, you'll need to use quotes. Just to give you an idea, the sample program contains questions from nursery rhymes and fairy tales.

Adjust the DIMension statement for the number of questions you put in your DATA statements. For an example, I've used only 20 questions. You'll probably want many more for a trivia quiz. Also define N in line 120 for the number of possible questions.

After the computer chooses and prints a question, the player inputs the answer. The computer then prints the correct answer. The computer does not keep score, however, because you may want to accept several forms of the answer. For instance, you could accept the answer to the question of what Little Miss Muffet ate as "curds and whey," "her curds and whey,"or even "cottage cheese"—but you would not accept "porridge."

Trivia Quiz simply asks the questions, so any number of people can play. You may want to expand the program to ask questions of each player in turn, to add scoring, or to make a more complex game. Of course, you can add graphics and sound, too.

If you wish to save typing effort, send a blank cassette or disk, a stamped, self-addressed mailer, and $3 to:

C. Regena
P.O. Box 1502
Cedar City, UT 84720

Please be sure to specify that you want the TI version of Trivia Quiz.

Answers To Reader Questions

I cover TI BASIC exclusively in this column because most COMPUTE! readers have console BASIC only. For other programming languages and hardware information, a good newsletter is published by the 99'ers User Group Association, 3535 South H Street, #93, Bakersfield, CA 93304. This is a nonprofit organization that communicates with hundreds of local user groups.

User groups are still going strong and are your best source of information and help. Many publish excellent newsletters (sorry, too numerous to mention here, and I wouldn't want to slight anyone).

New software is still being produced for the TI—I guess because there are over two million TIs out there. Most new programs are entertainment and educational titles. And yes, the Extended BASIC module remains available.

Hardware also is abundant. There are separate units for each peripheral, which saves you money if you need only one item. There are also combinations. One attractive unit I've used is CorComp's expansion unit that contains an RS-232 interface, 32K memory expansion, and a disk controller, all in a box about six inches wide that connects to the side of the console. It works just like the TI Peripheral Expansion Box but without the bulk and the noisy fan. My only complaint is that the disk drive connector isn't the same as the TI disk drive ribbon cable, but I understand Radio Shack has the necessary connections.

Next month I'll discuss how to use peripherals in your programming.

Trivia Quiz

```
100 REM  TRIVIA QUIZ
110 DIM S$(20),A$(20)
120 N=20
130 CALL CLEAR
140 PRINT TAB(8);"TRIVIA QU
    IZ"
150 PRINT ::"A QUESTION WIL
    L BE SHOWN."
160 PRINT ::"TYPE THE ANSWE
    R (WITHOUT"
170 PRINT :"COMMAS) THEN PR
    ESS <ENTER>."
180 PRINT ::"THE CORRECT AN
    SWER IS SHOWN."
190 PRINT ::"PRESS THE SPAC
    E BAR TO"
200 PRINT :"CONTINUE."
210 FOR C=1 TO N
220 READ S$(C),A$(C)
230 NEXT C
```

```
240 PRINT :::"PRESS ANY KEY
    TO START."
250 CALL KEY(0,K,S)
260 IF S<1 THEN 250
270 FOR C=1 TO N
280 CALL CLEAR
290 RANDOMIZE
300 R=INT(N*RND)+1
310 IF S$(R)="" THEN 300
320 PRINT S$(R)::
330 CALL SOUND(100,1497,2)
340 INPUT B$
350 PRINT :A$(R)
360 CALL KEY(0,K,S)
370 IF K<>32 THEN 360
380 S$(R)=""
390 NEXT C
400 CALL CLEAR
410 REM  PUT QUESTIONS HERE

420 DATA WHO LOST THEIR MIT
    TENS?,THE THREE LITTLE
    KITTENS
430 DATA WHO WERE THE THREE
     MEN IN A TUB?,"THE BUT
    CHER, THE BAKER, THE CA
    NDLESTICK MAKER"
440 DATA WHEN WILL THE CRAD
    LE ROCK?,WHEN THE WIND
    BLOWS
450 DATA WHAT DID JACK AND
    JILL(6 SPACES)FETCH?,A
    PAIL OF WATER
460 DATA WHO CUT OFF MICE'S
     TAILS?,THE FARMER'S WI
    FE
470 DATA HOW MAY BAGS OF WO
    OL DID THEBLACK SHEEP H
    AVE?,THREE
480 DATA WHO VISITED THE TH
    REE BEARS'HOME?,GOLDILO
    CKS
490 DATA WHO CHOKED ON AN A
    PPLE?,SNOW WHITE
500 DATA WHO SANG FOR HIS S
    UPPER?,LITTLE TOMMY TUC
    KER
510 DATA WHAT DID THE THREE
     PIGS USE TO BUILD THEI
    R HOUSES?,"STRAW, STICK
    S, BRICKS"
520 DATA WHO USED HER LONG
    HAIR TO(3 SPACES)SEE TH
    E PRINCE?,RAPUNZEL
530 DATA WHAT DID JACK FIND
     IN HIS(3 SPACES)CHRIST
    MAS PIE?,A PLUM
540 DATA WHAT DID LITTLE MI
    SS MUFFET EAT?,CURDS AN
    D WHEY
550 DATA WHO WAS LITTLE RED
     RIDING(3 SPACES)HOOD G
    OING TO VISIT?,HER GRAN
    DMOTHER
560 DATA WHAT COULDN'T JACK
     SPRAT(4 SPACES)EAT?,FA
    T
570 DATA WHO STOLE A PIG AN
    D AWAY QUORUN?,"TOM, TH
    E PIPER'S SON"
580 DATA WHO JUMPED OVER A
    (11 SPACES)CANDLESTICK?,
    JACK
590 DATA WHO SLEPT WITH HIS
     STOCKINGSON?,MY SON JO
    HN
600 DATA WHOSE FOOT FIT THE
     GLASS(4 SPACES)SLIPPER
    ?,CINDERELLA
610 DATA WHAT KIND OF MEAT
    DID ONE OFTHE LITTLE PI
    GGIES HAVE?,ROAST BEEF
620 END
```

©

TI Webster Dines Out

In line 480 of this game program from the June issue (Program 6, p. 57), the DISPLAY AT(3,22) should be DISPLAY AT(23,22). Reader Andrew Sonon supplied this correction, which moves the score indication to its proper place on the screen.

Apple MLX Error Messages

Although there are no errors in the "Apple MLX" listing from the June issue (p. 114), a number of readers have encountered DISK ERROR messages at unusual times. During normal operation of the program, the only errors that should occur are ones involving disk access; line 100 traps these errors. However, a side effect of this error trapping is that typing mistakes you make while entering MLX can also produce the message in line 610. Thus, if MLX gives you a DISK ERROR message when you are not accessing the disk, you need to check for a typing mistake in the MLX program. Lines 330–340 are a common problem area. Make sure you have not confused the letter O (used in the variable names O$ and O in those lines) with the number zero, which also appears in line 340. In COMPUTE!'s listings, a zero always has a diagonal slash through it.

Commodore Disk Editor

The POKE 1024, PEEK(254) in line 260 of this disk utility program from the June issue (p. 98) prevents you from changing the value of the first byte in a sector. This is the track number for the next sector in the file, so you may not need to change its value very often. However, you can modify the program to allow editing of the first byte by replacing the GOTO 260 at the end of line 310 with WAIT 198,255: GOTO 280.

Editing Enhancement For Softball Statistics

This record-keeping program from the July issue (p. 30) works as published for all the listed computers. However, the data input process can be simplified by allowing corrections for each player's statistics. To accomplish this, make the following changes to the general program (Program 1):

```
545 PRINT
546 PRINT "EVERYTHING OK (Y/N)
    ?"
547 INPUT A$
548 IF A$<>"N" AND A$<>"Y" THE
    N 545
549 IF A$="N" THEN 420
```

TI-99/4A users should also make the following additional changes:

```
548 IF (A$<>"N")*(A$<>"Y") THE
    N 545
```

Atari users should also make the following additional changes:

```
250 NEXT I:PRINT "{DOWN}E
    verything OK (Y/N)?":
    GOSUB 630:IF A$="N" T
    HEN 210
255 GOSUB 460
```

Mindbusters For Atari DOS 3.0

To use the Atari version of this thinking game from the April issue (p. 50) with DOS 3.0, you must delete the DIM K(255) in line 5 and make the following changes to line 1:

```
1 DIM K(255):FOR I = 0 TO 255: K(I) = 0:
  NEXT I: GOTO 5
```

©

Archon II: Adept

Arthur Leyenberger

Requirements: Commodore 64 or 128; Atari 400/800 or XL with at least 48K RAM; or an Apple II-series computer with at least 48K RAM. All versions also require a disk drive and a joystick.

In any entertainment business, whether it is movies, books, or videogames, there is a natural tendency to produce sequels to existing hits. Making sequels can be approached in a number of ways. Often the sequel is just more of the same thing, such as *Jaws II* and *Jaws 3-D*; the hope is that there will be continued demand for more of the same thing. The risk in this approach is that people may eventually grow tired of the old formula.

Another approach is to use the same basic theme of the original, but add something new or better to the sequel—as in the successful *Star Wars* and *Star Trek* films. This is also the case with *Archon II: Adept*, a new game from Free Fall Associates, published by Electronic Arts.

Adept was designed to be a game that has the same mixture of strategy, action, and play mechanics as the original *Archon*. But it is sufficiently different to please both new players and long-time *Archon* devotees.

Casting Magic Spells

Adept is basically a game of magic, focusing on energy and resource management. Unlike *Archon*, it allows people of different skill levels to compete more equally. The combination of strategic board play and individual combat means that people who don't have fantastic reflexes have a reasonable chance of winning.

Each side starts with four Adepts, similar to the Wizard and the Sorceress pieces in *Archon*. The game begins with one Adept on each of the four elements: Earth, Water, Air, and Fire. The elements are represented by colored bands on the screen. Your pieces are shown vertically on each side of the screen, with the more powerful, flexible pieces at the top.

The more powerful pieces require more energy to manipulate. Choosing the beginner level gives you the most energy while choosing the advanced level gives you the least. The play level also affects the speed of the pieces in combat. Each piece can cast spells, and every spell costs magical energy. You can cast as many spells as you want with any piece as long as you can afford it. To gain magical energy, you must occupy "power points." If you occupy all of the power points, you win the game.

You shape your magical army depending on such factors as your skill with a particular piece, a certain strategy, or the pieces you like. Each side has four different elemental pieces that can be called upon. They are strongest in their own element but can be played in any element. Some of the pieces are common to both sides, while others are unique.

When you choose a piece, you are shown the amount of energy you currently have as well as how much energy it costs to use that piece or cast a spell. Although many of the spells are familiar to the experienced *Archon* player, some have new twists. For example, the Imprison spell lets you trap an enemy piece as long as you have the energy to pay for it. Casting and maintaining this spell costs energy during every turn, so imprisoning everyone in sight would soon drain your resources and lose you the game.

Apocalypse Now

All of the spells can be cast repeatedly, with one exception: the Apocalypse. You can cast this spell only once, since it begins a battle that ends the game. It is a one-on-one, winner-take-all battle that is shaped by your strategic position: the amount of energy, number of pieces, and surviving Adepts you have left. The result can be either a well-matched or very one-sided battle.

This go-for-broke spell typically is cast in two situations. One is if you are way ahead and, for some reason, are having trouble occupying the last power point. The other situation is when you're in danger of losing the game and have no other way out. Since the Apocalypse spell is expensive, casting it in a weak position weakens you still more.

Adept contains a wider variety of creatures than are found in *Archon*. Each piece has a unique weapon and performs best in a certain element. For example, the Juggernaut is best suited to Air and uses itself as a missile. When fired, it glows and charges across the screen. While in motion, it is invulnerable and destroys anything in its path. Using the Juggernaut in the water, however, significantly slows it down and makes it more vulnerable to attack.

In 1983, *Archon* ranked at the top of almost every gamer's list. It still belongs in the videogame hall of fame. *Archon II: Adept* is even better than *Archon* and should rank even higher with experienced *Archon* devotees as well as with players new to the world of magic and strategy.

Archon II:Adept
Electronic Arts
2755 Campus Drive
San Mateo, CA 94403
$33 (Commodore & Atari)
$40 (Apple)

WordPerfect For IBM

Richard Mansfield, Senior Editor

Requirements: IBM PC or compatible with at least 192K of RAM, two disk drives, DOS 2.0 or higher, and a printer. Not compatible with the PCjr.

This is an excellent word processor. In addition to performing all the usual tasks with speed and efficiency, *WordPerfect* includes many features which are either rare or unique.

Perhaps its best single feature is that it gives you a blank screen to write on—no distractions from control codes, command lines, or other clutter. Just a couple of unobtrusive numbers in the lower-right corner to identify the col-

DISK WORLD!
is pleased to announce the *lowest*-prices ever on brand name diskettes!

You've seen ads for lower priced diskettes.

But did you ever notice that they usually don't carry or list a brand name. There's a reason for that: there is no brand name to be listed, since these low-cost "generic" diskettes are politely called garbage within the diskette industry.

One or two individuals buy reject product from lower-quality manufacturers and put it out in the market place without a label...and without any meaningful guaranty that you're getting good product.

But not at DISK WORLD!

We deal only with **branded** merchandise...with diskettes made by people who are proud to have their name known.

3M: Almost every diskette you buy has a little 3M in it. 3M manufactures more blank diskette media than anyone else...and probably produces more computer media in all forms than any other manufacturer. This is factory- fresh 3M product, boxed in 10's, reinforced hubs, user ID labels, write - protect tabs and a **100% LIFETIME WARRANTY.**

5.25" SSDD w/ **FREE** Flip 'n File 1S.....$ 1.42 ea.
5.25" DSDD w/ **FREE** Flip 'n File 1S.....$ 1.74 ea.
(Above FREE offers subject to limited availability.)
5.25" SSDD-96TPI.....$ 2.06 ea.
5.25" DSDD-96TPI.....$ 2.57 ea.
5.25" DSDD-HD for IBM PC/AT.....$ 3.93 ea.
3.50" SSDD-135TPI for Mac and others.....$ 2.86 ea.

BASF: Exceptional value from one of the world's largest companies. Except where indicated, BASF diskettes are packaged in cardboard boxes of 10, reinforced hubs, Tyvec sleeves, user ID labels and write - protect labels. Of course, there's a **100% LIFETIME WARRANTY** on all BASF product.

5.25" SSDD.....$ 1.14 ea.
5.25" DSDD.....$ 1.31 ea.
5.25 DSDD-HD for IBM PC/AT.....$ 3.14 ea.
5.25 SSDD in Bulk Packs of 50.....$ .89 ea.
5.25" DSDD in Bulk Packs of 50.....$ 1.12 ea.
5.25" "FLIPPY" diskettes ub Bulk Packs of 50.....$ 1.20 ea.
3.50" SSDD-135 MICROFLOPPY for Macintosh, others.....$ 2.34 ea.

ATHANA: The diskette you've used hundreds of times...without knowing it.

ATHANA edalizes in producing top-of-the-line diskettes for software ea.manufacturers . CornS piles with Tyvec sleeves, user ID labels, reinforced hubs and write-protect tabs. **100% LIFETIME WARRANTY.**

5.25" ISSDDn Bulk Packs of 50.....$ .97
S.25 DSDD in Bulk Packs of 50.....$ 1.01 ea.

NASHUA: One of the largest diskette manufacturers in the nation.....and the leader ir rigid disk packs for maintframes. . Of course, there's a **100% LIFETIME WARRANTY** on all NASHUA products.Tyves sleeves, reinforced hubs, user ID labels and write - protect tabs are included.

5.25" SSDD Bulk Pack in 50's.....$ .97 ea.
S.25 DSDD Bulk Pack in 50's.....$ 1.07 ea.

MEMOREX: Is it real or is it Memorex? It's Memorex, packaged as shown, complete with Tyvec sleeves, reinforced hubs, user ID labels and write - protect †abs. And , of course,a **100% LIFETIME WARRANTY>**

5.25" SSDD in boxes of 10.....$ 1.27 ea.
5.25" DSDD in boxes of 10.....$ 1.69 ea.
5.25" SSDD "FLIPPY" diskettes in boxes of 10.....$ 1.59 ea.
5.25" DSDD-HD for IBM PC/AT in boxes of 10.....$ 3.86 ea.
3.50 SSDD-135TPI MICROFLOPPY in boxes of 10.....$ 2.42 ea.

SONY: The company that invented the 3.50" drive...and licenses it to every other manufacturer. Boxed in 10's with user ID labels. (There's no need for write-protect tabs or Tyvec sleeves.

3.50" SSDD for the Macintosh.....$ 2.55 ea.
5.25" DSDD for Data General/One and HP.....$ 3.89 ea.

RIBBONS:

Epson MX-70/80.....$ 3.65 ea.
Epson MX/FX-100.$ 4.9S ea.
Okidata Micro83.....$ 1.S1 ea
Okidata Micro84.....$ 3.73 ea.
Add .25 each ribbon for shono.

DISKETTE STORAGE:

Amaray 50.....$ 9.69 ea. + $ 3.00 Shpng.
Amaray 30.....$ 9.22 ea., +$ 2.00 Shpng.
Diskette 70.....$ 9.9S. + $ 3.00 Shpng.
Disk Kaddies....$ 1.6S ea. + .25 Shpng.

ORDERING INSTRUCTIONS: Add $ 3.00 shipping per 100 diskettes or fraction thereof. All other items, add shipping as shown. VISA, MASTERCARD AND PREPAID orders accepted , COD orders, add $ 5.00 Special Handling fee. APO, FPO, HI, AK and PR orders, add 5% to cover Parcel Post shipping. No foreign orders. Ill. residents, add 7% sales tax.MINIMUM ORDER: $ 35.00 or 30 diskettes.

DISK WORLD!, Inc.
(In Illinois 1-312-256-7140)

TDLL FREE: 1-800-621-6827
629 Green Bay Rd. Wilmette , Illinois 60091

HOW TO TURN YOUR PERSONAL COMPUTER INTO YOUR PERSONAL BANKER.

Here's how to make the investment in your personal computer really pay off.

With SPECTRUM,℠ the new electronic home banking and information system from The Chase Manhattan Bank, N.A. Teamed up with your PC, SPECTRUM is your direct link to Chase. And the start of a better way to manage your money and your time.

A push of a button lets you pay bills electronically, anywhere; transfer funds; keep records, and more. Even get vital financial information and trade stocks* at discount rates.

Right at home, anytime –with complete security. All, including electronic mail, starting at just $5 a month. And with our new subscriber rebates and free offers, now's an even better time to get control of your finances. Call today for your free demo diskette.

CALL TOLL FREE
1-800-522-7766

CHASE

© 1985 The Chase Manhattan Corporation.*Stocks are traded through Rose & Company,a Chase affiliate.

umn, line, and page. You can call up a screen with control codes, but when you're writing, what you see is what you get: Underlined passages are underlined, bold is bold, indenting looks like it will on paper.

The spelling checker is another extraordinary feature; it's fast and effectively organized. The dictionary comes with 100,000 words (more than most paperback dictionaries), is easily expanded, stunningly quick, and simple to use. It even looks up words phonetically: If you type *komitty* it will look up *committee* for you. It checks for double words (such as "Paris in the the spring") and gives a word count as well.

If there are things you do repeatedly, you can create macros, little programs that control the word processor. Then, at the touch of a single key, you can type in a frequently used address, save the file, search for another file, etc. You can also create two screens and switch between them, and instruct the word processor to automatically save backup copies of your current work every few minutes. And *WordPerfect* has a sophisticated, efficient footnote capability.

Sophisticated and efficient are good words to describe this program. While it is easy to use, it contains so many features and advanced commands that even the most demanding writers should find what they need. Statistical typing, math calculation, automated outline numbering, automated compilation of tables of contents—the list goes on. If you need some arcane function, you're likely to find it in *WordPerfect*. If you simply need a powerful, sensible writing machine, you owe it to yourself to consider this fine tool.

WordPerfect
SSI Software
325 North State Street
Orem, UT 84057
$495

Adventures In Narnia For Apple And 64

C. Regena

Requirements: Commodore 64 with a disk drive; or an Apple II-series computer with at least 48K RAM and a disk drive.

Adventures in Narnia is an adventure game based on the book *The Chronicles of Narnia* by C.S. Lewis. When we first opened the package, my son exclaimed, "Hey, I've read that book!" Inside is a paperback entitled *The Lion, the Witch and the Wardrobe*, a popular story in the Chronicles series which was the inspiration for *Adventures in Narnia*. The characters in the game are the same as those in the book. You don't have to read the book to play the game, but if you play the game and enjoy it, you'll probably want to read the book.

Adventures in Narnia is actually a combination adventure, arcade, and board game. It even comes with a deck of game cards and a pair of dice. High-resolution computer graphics replace the traditional board, but sometimes during the arcade action you bump into something that requires you to roll the dice or choose a card. *Adventures in Narnia* was designed to use the computer, but still resemble a board game and bring human interaction into play. As a result, the game is not always machine-controlled. Its authors point out that the computer waits while you "think, discuss, decide." You can "strategize and argue (without penalty) in the middle of the game, allowing the fun and humor of dialogue that is missing in normal videogames."

Other adventures are available in this series as well—such as *Dawn Treader*, based on the story *The Voyage of the "Dawn Treader."*

Dodging Evil Dwarfs

The game actually is a two-part adventure with two different arcade screens. In the first adventure, you start out in the wardrobe and try to gain points and strength. In the second adventure, you use the strength to reach a character called Aslan the Lion.

You start the game by shuffling the cards and placing them near the computer. Your player is at the right side of the screen among randomly placed bushes, flowers, and beavers. Evil dwarfs dart about, and you have to avoid them. Dwarfs can also hide in the bushes, so you don't want to hit a bush. You can gather flowers to gain points, and you can meet a friendly beaver to gain strengths (indicated by hearts at the top of the screen). The evil dwarfs patrol Narnia and go around stomping on flowers and beavers to prevent your success.

The game action is quick. If you hit a dwarf (or a moving bush), you're sent to a dwarf battle. The dwarf thinks of a random number, and you must roll the dice to beat his number. If you win, you get 500 points, but if you lose, you sacrifice one strength.

From time to time, Edmund and the Witch appear on the screen. Your job is to intercept Edmund to prevent him from reaching the Witch. The first adventure ends if the Witch captures Edmund, or if you get ten heart strengths, or if your time runs out. Then the next adventure starts.

Inside The Ice Maze

The second adventure takes place in the Ice Maze with your character at the lower-right corner of the screen and Aslan the Lion at the upper left. Your object is to get to Aslan, but the Witch sends evil crystals through the maze to stop you. If you get hit by a crystal, you're sent back to the beginning and you lose one heart. If you lose all the hearts and get hit again, you lose the game.

The gravity chutes are another hazard. Snow is falling inside these chutes, and if you step into one, you'll tumble to the bottom and find yourself sealed inside the maze.

If you run into a door, you're sidetracked to a subadventure. Since the game action is fast, you'll probably run into some doors by accident. Once in a subadventure, you're directed to pick a card. Then you enter the card's code into the computer. If your card is Aslan the Lion, you automatically gain one heart strength. If you draw a Zap card, you're automatically sent back to the beginning of the maze and you lose one strength. Other cards—such as Fenris Ulf the Wolf, Cair Paravel the Castle, and Jadis the Witch—require you to roll the dice to determine your consequences, which can be good or bad. After the subadventure, you return to the maze for another crack at Aslan the Lion.

You lose the game if the Witch steals all your hearts with her ice crystals or if time runs out. You win the game by reaching Aslan the Lion. Your final score is determined by the running score on the screen plus 1000 points for foiling the Witch, 1000 points for each remaining heart, and the remaining time multiplied by 10.

The instruction booklet that comes with this package is very good. It presents all aspects of the game so you can identify each object and recognize whether it is good or bad. Color screen photos are accompanied with explanations for each possibility. And as you play the game, the screen instructions also are easy to understand.

If you own more than one computer, note that the Apple and 64 versions of this program are on flip sides of the same disk.

Adventures in Narnia
Word, Inc.
4800 W. Waco Drive
Waco, TX 76796
$39.95

Classified

COMPUTE! Classified is a low-cost way to tell over 350,000 microcomputer owners about your product or service.

Rates: $25 per line, minimum of four lines. Any or all of the first line set in capital letters at no charge. Add $15 per line for boldface words, or $50 for the entire ad set in boldface (any number of lines.)

Terms: Prepayment is required. Check, money order, American Express, Visa, or MasterCard is accepted.

Form: Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words. Please underline words to be set in boldface.

General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Orders will not be acknowledged. Ad will appear in next available issue after receipt.

Closing: 10th of the third month preceding cover date (e.g., June issue closes March 10th). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.

Notice: COMPUTE! Publications cannot be responsible for offers or claims of advertisers, but will attempt to screen out misleading or questionable copy.

SOFTWARE

TI99 LOTTO PICKER. Go for Million Dollar Jackpots! Picks LOTTO, Daily Numbers, & Win-4. All USA Lotto games are programmed in. $25. RIDGE, 170 Broadway, #201-C, NYC, NY 10038. 718-833-6335. Free Catalog.

IBM PCjr OWNERS: We have the software you want. We also have kits for easy 2nd drive addition. Free catalog. OOWL SOFTWARE, 1435 Burnley Sq. N., Columbus, OH 43229.

PRINT USR - Formatted numeric output. Use with your BASIC program. ML speed. 48K+ Apple or C64. Send $9.95 to: MAK Ent., 1702 W. Lender, Mesa, AZ 85202.

AMCUP, AN ORIGINAL sailboat racing game. 1 or 2 players. C-64 w/1541 + joystick. $14.95 ch./m.o. SASE for info. Dream Dog Software, P.O. Box 1724, Eugene, OR 97440.

"CRAPSMAN!" The C-64 casino craps tutor! Learn all bets. Test systems. Up to four players. $31.95 pp, disk. EMCEE SOFT CORP., P.O. Box 402124, Miami Beach, FL 33140.

SHOPPING LIST PROGRAM for C-64 with 1541 disk and printer. It prepares a shopping list for meals you select, organized by your market sections. Send $42 to 4955 Woodman Ave., Sherman Oaks, CA 91423

SOFTWARE PROTECTOR 64. Copy and list protect your valuable Basic programs. Only $10 (disk only). Systems Software, 44 Rte 25A #209, Smithtown, NY 11787.

PINOCHLE for Commodore 64, PC and PCjr. Real, double-deck, partnership pinochle gives you a partner and two opponents! On disk, $25 postpaid. Jim Bernard, 8018 Sunset Path Ct., Springfield, VA 22153.

TI CHECKWRITER Letter code and dollar amount are all that's necessary to print a complete check. Provides yearly totals for taxes. Disk $24.95. R.I.P. Software 248 Purisima Rd., Woodside CA 94062.

SCIENCE SOFTWARE with graphics for the Commodore 64. Astronomy programs, etc. Catalog and demo disk $3. David Eagle, 7952 W. Quarto Dr, Littleton, CO 80123.

FIXED ASSET DEPRECIATION on C64. Handles acrs and pre-acrs methods. $39.95. For free info write: MPM Software Prods., P.O. Box 3522, Glendale, CA 91201.

ATARI USERS. 4 games on cassette, $25. Saucers + monsters, poker, sea finder, pinball. Write or call Keith Anderson, 8435 12th Ave., Silver Spring, MD 20903 (301) 434-0285.

HARDWARE

REPLACEMENT POWER SUPPLIES FOR C-64 or VIC-20. $29 ea. plus $3 shipping. 64 parts list $5-refundable w/1st order. AA Computer, 2726 Park St., JAX, FL 32205.

NUMERIC KEYPAD with joystick CURSOR control, gold contact $75. Centronics interface $40. Autodos cartridge automatically loads programs from easy-to-use menu $40. All C-64 fully compatible. SS 64 RF Mod. $40. Battery Back-up with best available protection for C-64 $50, Apple/IBM $150 (U.S.) APPLIED COMPUTING, Box 1566, PTBO, ONT. K9J 7H7. Send $2 for cat. Call (705) 745-8617 anytime.

MISCELLANEOUS

Trade in your used Commodore or Atari on a brand new C-128 or Atari ST. This offer may not be available through retail outlets. Brochure $1.50 and SASE. NEW WEST TECHNOLOGY, 4B Monroe Pkw., Box 200, Ste. 134, Lake Oswego, OR 97034

FLYING COMPUTERS but stuck on the ground? Learn requirements to fly aircraft, with or without license. $3.00 to: Aviation Resource, P.O. Box 369-C, Lancaster, NY 14086.

Super 99 Monthly—quality info for TI-99/4A & compatibles since 09/84. US $12/yr. Other $16/yr. Bytemaster Computer Svces., 171 Mustang, Sulphur, LA 70663.

FREE! SHOP BY MODEM in our revolutionary electronic shopping mall. We even accept credit cards! We have products YOU want! Call 1-818-840-8066 with your modem NOW!

BUY-SELL-TRADE computers, hardware, software. Announce clubs, BBS's, etc. Send for free sample of Micro-Swap, the computer classifieds. P.O. Box 24, Esmond, IL 60129.

COMPUTER OWNERS!!! Earn dollars and save money with your computer. Write C&D Associates, Box 851, Mt. Prospect, IL 60056. Specify computer type. No Investment!!

TI-99/4A Software/Hardware bargains. Hard to find items. Huge selection. Fast Service. Free Catalog. D.E.C., Box 690, Hicksville, NY 11801

LOWEST DISK PRICES-SS/DD with sleeve & label-10/$8.80, bulk 100/$78. Ds/DD with sleeve & label-10/$10.90, bulk 100/$99. Prime quality major mfr's overstock! Money-back satisfaction guarantee! Min order $15. Pay by MC/Visa/AE, UNITECH 20 Hurley St., Cambridge, MA 02141 (800) 343-0472, in Mass (617) 864-8324.

HELP IS ON THE WAY!
Just call 1-800-334-0868 to get your free copy of the latest COMPUTE! Books Catalog! If you need help in getting information on all of the latest COMPUTE! book titles available plus all COMPUTE! backlist titles, call us today!

SAFEGUARD YOUR COMPUTER KEYBOARD from contaminants with OVERBYTE molded keyboard covers. $24.95 xparent & durable. Call (213) 866-2583/send $2 further info. OVERBYTE-P.O. Box 10652, Burbank, CA 91510.

PHONEMARK DATASETTES: C64, Plus4, Vic20. $25.00 plus shipping. We are direct importers. American Mercantile Co., 2450 1st Ave., South, Seattle, WA 98134 (206) 624-6141.

DISK SERVICE MANUAL. Service floppies without special software or equipment. FREE information. Consumertronics-DSM, P.O. Drawer 537, Alamogordo, NM 88310.

MAXELL MD1, $1.39 - MD2, $1.99. Dysan 104/1D, $1.79 - 104/2D, $2.39. Shipping $3.75. Also Verbatim, IBM, 3M, BASF. TAPE WORLD, 220 Spring St., Butler, PA 16001, 1-800-245-6000, Visa, MC.

$$ MONEY MAKER $$
Increase your income by hundreds and even thousands of dollars. Money back guarantee! (Apple, Macintosh, IBM) Free brochure: 1-800-223-5838 (305-771-6498).

ANIMATOR

Steve Johnson

This feature-packed utility makes it a breeze to create your own short cartoons or animation sequences on the computer screen. The original version is for the IBM PC with BASICA and color/graphics adapter, or Enhanced Model PCjr with Cartridge BASIC. We've added versions for the Atari 400/800, XL, and XE with at least 32K RAM for disk or 24K RAM for tape; Commodore 64 and 128 (in 64 mode); TI-99/4A with Extended BASIC; and Apple II-series computers with at least 48K RAM. The Atari and 64 versions also require a joystick.

Computer animation can be marvelous to behold but a drudge to produce. Whether you're working in BASIC or machine language, creating objects and manipulating them on the screen can mean fumbling for hours with PEEKs, POKEs, bits, bytes, and other tedious details.

"Animator" goes a long way toward automating this process. It works much like a cartoonist's sketchpad, letting you draw a series of similar images which are then displayed in rapid sequence to create the illusion of movement. Your finished cartoons can be saved on disk or tape and reloaded for viewing later.

What every Apple II+ and IIe user should ask before buying the "Sider" 10 MB hard disk:

When a company offers a superior quality 10 megabyte Winchester hard disk for only $695, it's bound to raise a few eyebrows...and a lot of questions. The fact is, you're probably already wondering "Can I really get a 10 megabyte hard disk that's *reliable* for only $695?" The answer is: ABSOLUTELY...when you choose the Sider from First Class Peripherals.

What's so great about the Sider?

For starters, the Sider lets you boot your ***Apple II+*** or ***IIe*** directly off the hard disk—unlike some other Winchester subsystems. Rebooting is also trouble-free. And the disk is partitionable, allowing you to allocate space to four operating systems on the same disk. The Sider supports: Apple DOS 3.3; Pro DOS™; Apple Pascal; and CP/M®

What's more, a small "footprint" lets you incorporate the compact Sider into your existing computer set-up with ease.

In addition, with the Sider, you not only pay far less for the subsystem, you also save money on installation. Because, unlike other 10 MB systems that require the purchase of expensive "extras," the Sider is *plug and play*. Everything you need is provided, including cables, host adaptor, installation software and manual.

What makes it so reliable?

To start, the Sider is manufactured, and sold exclusively, by First Class Peripherals, an innovative computer company which is backed by Xebec. The computer industry's leading manufacturer of disk controllers, Xebec has over a decade of experience serving customers like IBM, Toshiba, Texas Instruments and Hewlett Packard. It's this kind of expertise that helps assure the Sider's performance.

Special design features further enhance reliability. The Sider's controller is the field-proven, industry standard Xebec S1410A. And Xebec's 3200 drive tester, the toughest in the industry, ensures that the Sider will operate reliably. One more assurance of the $695 Sider's quality: it's UL Approved and FCC Class B rated.

"Only $695?"

But why is it only $695?

You pay less for the Sider than for other 10 MB hard disks simply because you're paying for the superior quality components inside the unit, not for a lot of retail overhead costs. Since First Class Peripherals sells direct, you avoid dealer and distribution expenses, and pay only for the product.

What about a guarantee?

Like many experienced Apple users, you may be reluctant to buy a hard disk priced at only $695 without first seeing for yourself how it performs. That's why First Class Peripherals offers you a reassuring, money-back guarantee that eliminates any risk on your part. Simply order the Sider and use it for 15 days. Then, if you're not entirely satisfied, return it and receive a full refund—no questions asked.

The Sider also comes with a full one-year limited warranty. Plus, there's a convenient, toll-free hotline you can call anytime you have a technical or service question, or need help.

Don't delay. Order the Sider now.

To receive the Sider 10 megabyte Winchester hard disk subsystem for only $695, simply order using the coupon below. For faster service, order by phone and charge to your VISA, MasterCard or American Express. (You can also call us if you have any questions or technical concerns about the Sider. We'll see you get the help you need.) Call toll-free:

1 800 538-1307
Extension 709

☐ Yes, please send me the Sider, including half-height 10 megabyte Winchester hard disk drive, Apple adaptor board, cable, complete installation software and documentation.

I prefer to pay as follows:

☐ I've enclosed my check or money order for $695* + $15 shipping and handling, payable to First Class Peripherals.

☐ Please bill the following credit card account for $695* + $15 shipping and handling:

☐ VISA ☐ MasterCard ☐ American Express

Card # *Exp. Date*

Signature

Name

Address

City

State *Zip*

Telephone (area code)

Mail to:

FIRST PERIPHERALS CLASS
3579 Highway 50 East
Carson City, NV 89701

*Residents of CA, NV and PA, please add appropriate sales tax.

If you're using the IBM PC/PCjr version of Animator, type in and save Program 1, then run it and follow the instructions below. *If you have an Atari, 64, TI, or Apple, you should also read the following instructions as a general guide to using Animator.* Then refer to the typing instructions and programming notes specific to your computer for additional details.

Drawing An Image

When you run Animator, it displays an editing screen with 20 numbered frames. You can draw as many as 20 pictures, one in each frame, then flip rapidly through the frames to create animation. The frame number displayed at the upper left of the screen shows which frame you're currently working on. Normally, Animator begins the animation with frame 1 and ends with frame 20. But you can start and end the animation wherever you like. For example, a short sequence might start with frame 1 and end with frame 3. To view only part of a long sequence, you might start at frame 12 and end at frame 18, and so on. The frame number is controlled by pressing the right and left arrow keys.

The frame number also determines which frame you'll be working on when you go to the editing screen. Let's start with a simple example. Make sure the frame number is set to 1, then press the 2 key to select the editing function and press Enter at the next prompt. After a brief pause, Animator displays a drawing grid with a blinking cursor. Edit mode has three main functions, selected by pressing different keys. Press D to draw with the cursor, E to erase, and M to move the cursor without disturbing anything on the screen.

Draw a simple shape on the grid to become familiar with these basic functions. As you'll see, Animator displays the shape in its actual size to the left of the drawing grid. An inverse function lets you reverse everything on the grid—every dot becomes a blank, and vice versa (be patient—it takes Animator about a minute to complete this process).

Once the picture is finished, you can press S to save it and return to the main screen. *Note that you must save a picture with S to put it in the frame.* If you exit the edit mode by pressing Q, the new picture is lost and Animator uses whatever that frame previously contained. Try drawing a simple shape and saving it with S (since this is just for practice, any scribble will do). When you return to the main screen, Animator displays the picture in frame 1.

Frame By Frame

Now you're ready to draw the next frame in the sequence. In most cases you'll want to make only slight changes from one frame to the next, to simulate smooth motion. To save time, Animator lets you copy a picture from one frame to another. Let's demonstrate this by copying the picture from frame 1 to frame 2. Set the picture number to 1 with the arrow keys, then press 2 to edit. Animator displays a prompt, inviting you to enter a frame number. To edit the current picture number, you would just press Enter. However, by entering a *different* number you can copy the current picture into a different frame, then change that picture to make the next frame in your cartoon.

When you enter 2 at the prompt, Animator copies the picture from frame 1 into the drawing grid. When the drawing grid appears, make some change in the picture to distinguish it from frame 1. Now press S to save the picture in frame 2 and return to the main screen. Animator displays both pictures in their respective frames.

After drawing a few frames, you're ready to bring them to life. The first step is to specify the starting and ending frame numbers. The starting number determines which frame begins the animation, and the ending number tells Animator where the series ends.

Set the starting number first. Use the arrow keys to set the frame number to 1, then press the 3 key. Now use the arrow keys to make the frame number match the *last* frame that contains a picture, then press the 4 key. This sets the ending number. You must always set the starting and ending numbers before selecting animation (if you don't, Animator flips through all 20 frames whether they contain pictures or not). Once these numbers are set, press the 1 key to view the sequence. Press the space bar to pause and Enter to stop it.

By selecting different *speed* and *pause* values, you can move the animated figure across the screen. The speed value can range from −15 to 15. When it is 0, the figure is animated in place; positive values move the figure from left to right, and negative values move it from right to left. The greater the value, the faster the figure moves. Press the 5 key to decrease the animation speed, and 6 to increase it.

The pause value controls the time delay between each frame of the animation. A small pause value makes the pictures change very quickly, while larger values slow down the process.

Macro Editing Features

Animator provides a few macro (large-scale) editing features to help you work with longer cartoons. The insert function lets you insert a blank frame anywhere in the series. To use it, set the frame number to the number of the frame where you want to insert a blank, then press the I key. The designated picture (and all those following it) are bumped forward one frame. Note that the picture in frame 20 is always lost when you insert.

The delete function lets you delete any frame in the series. Change the picture number to the frame you want to eliminate, then press D. All the higher numbered

88K *Lowest Price In The USA!* 152K

ATARI®

Computer System Sale

• Students • Word Processing • Home • Business

[1] ***Rated "Best Buy" by Consumers Digest Buyers Guide, January 1985***

LOOK AT ALL YOU GET FOR ONLY **$449** SYSTEM PRICE

	LIST PRICE	INDIVIDUAL SALE PRICE
Atari 800XL 88K Camputer	$179.00	$109.00
Atari 1050 127K Disk Drive	299.00	199.00
Atari 1027 Letter Quality 20 CPS Printer	299.00	199.00
Atari Writer Ward Procesaer	59.95	39.95
Atari BASIC Tutarial Manual	16.95	12.95
All connecting cables & T.V. interface included. Monitors sold seporetly. **TOTALS**	**$852.90**	**$559.90**

SAVE $100
All 5 ONLY
$449.00
SYSTEM SALE PRICE

152K SYSTEM $499.00

Other Accessories	List	Sale
12" Hi Resolution Green or Amber Screen Monitor	$199.00	**69.95**
13" Hi Resolution Color Monitor	$399.00	**169.95**
* ATARI 130XE 152K Computer	$249.00	**159.00**

Add $9.95 for Connection Cables (Monitors Only)
(Ltd. Qty)
Add $10 for UPS

15 DAY FREE TRIAL We give you 15 days to try out this ATARI COMPUTER SYSTEM!! If it doesn t meet your expectations, just send it back to us prepaid and we will refund your purchase price!! **90 DAY IMMEDIATE REPLACEMENT WARRANTY**. If any of the ATARI COMPUTER SYSTEM equipment or programs fail due to faulty workmanship or material within 90 days of purchase we will replace it IMMEDIATELY with no service charge!!

Best Prices • Over 1000 Programs and 500 Accessories Available • Best Service • One Day Express Mail • Programming Knowledge • Technical Support

Add $25.00 for shipping and handling!!
Enclose Cashiers Check Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail! We accept Visa and MasterCard. We ship C.O.D. to continental U.S addresses only. Add $10 more if C.O.D.

COMPUTER DIRECT
We Love Our Customers
22292 N. Pepper Rd., Barrington, Ill. 60010
312/382-5050 to order

pictures move down one frame, deleting the picture in the designated frame. Frame 20 is always blank after a deletion.

The inverse function (press 9) works just like inverse in editing mode, but inverts all 20 frames at once.

To clear all 20 frames, press Q to quit or C to clear. Since these last two functions can have drastic results, Animator lets you abort either one without harm.

When you finish a sequence, press S to save it on disk. The screen clears and displays three options: You can Press A to abort the save, F to list the picture files on that disk, or any other key to continue with the save. Picture filenames are limited to eight characters (the first character cannot be a number). Do not add a three-character extension; Animator automatically appends the extension .ANI when you save or load a picture file.

Finally, Animator's program option (available *only* in the IBM PC/PCjr version) can write a separate BASIC program to display your cartoon. Press P to select this option, and sit back while Animator writes the new program to disk under the filename PRG.BAS. Afterward, Animator ends with a reminder to reload PRG.BAS and save it with a new filename. This prevents the program from being overwritten if you select this option again.

Commodore 64 Version

The 64 version of Animator is written entirely in machine language, but you can use it without understanding machine language at all. Type in and save Program 2 using the "MLX" machine language entry program printed elsewhere in this issue. Here is the information you'll need:

Starting address: 49152
Ending address: 52991

After you've saved "64 Animator," plug a joystick into port 2 and load the program with LOAD"*filename*",8,1 for disk or LOAD"*filename*",1,1 for tape. Type SYS 49152 and press RETURN to run the program.

64 Animator's main screen consists of 21 picture frames instead of 20 as found in the IBM version. It also uses sprites to animate the frames. Although the Commodore 64 normally is limited to displaying eight sprites at once, 64 Animator employs as many as 22 simultaneous sprites with the raster interrupt technique described in *COMPUTE!'s First Book of Commodore 64.*

When you begin the program, the frames may contain random data; Press C to clear them out. Most program functions are controlled with the joystick. Near the bottom of the screen you'll see a list of several functions (frame numbers, options, and so on, as described above in the general instructions). As you move the joystick up or down, the function you select is highlighted in reverse video. To increase or decrease the selected value, move the joystick left or right.

Press the E key to enter edit mode. The joystick moves the blinking cursor around the drawing grid, and the fire button toggles the space under the cursor on or off. To draw or erase more than one space at a time, hold the joystick button down and move the stick in the direction you want.

The current picture number is displayed to the right of the screen. You can move to a different picture within edit mode: Press the + key to increase the picture number, and the − key to decrease it. Animator always displays the current picture in actual size above the picture number. Above and to the right of the current picture is the next picture in the series, and above to the left is the previous picture. (If you haven't drawn any pictures yet, these frames may be blank or contain random shapes.)

Press I within edit mode to invert the shape (change dots to blanks, and vice versa). The cursor keys shift the entire shape one space inside the grid, either left, right, up, or down. You can also expand the picture horizontally (press X) or vertically (press Y). Expansion is toggled off by pressing the same keys, and can be used on the main screen as well.

Edit mode lets you copy a picture from one frame into another. Press the f7 function key to store the current shape in the picture buffer. Then change the picture number with + or − and press f8 (SHIFT-f7) to copy the picture from the buffer into the new frame. In this way you can quickly draw a series of shapes without leaving edit mode. The R key returns you from edit mode to the main screen.

Animating On The 64

Before watching the animation, set the starting and ending frame numbers to the appropriate range. The joystick button turns animation on and off. To change the speed during animation, move the joystick left or right. Press the f1 function key to clear everything but your animated figure off the screen (f1 also brings back the main screen). You can change the picture color by pressing the CTRL key and one of the number keys from 1–8. The colors are the same as those printed on the front of the keys. For example, press CTRL-0 to color every figure black.

Insertions and deletions work as in the IBM version, except that an insertion retains the original picture in the selected frame rather than inserting a blank. To quickly fill several frames with the same picture, press I several times.

64 Animator can save and load picture files with either disk or tape. Be sure to set the starting and ending numbers before saving, since nothing is saved outside that range. The prefix ANI. is added to picture files on disk; do not type this prefix when loading or saving. To abort a save or load, press RETURN.

It's possible to merge pictures from different files if the two files use different frame numbers. Simply load the second file after the first. The main screen now contains pictures from both files. If the files have overlapping frame numbers, the second file has priority. For example, say that you load ANI.A, which uses frames 1–3, then load ANI.B, which uses frames 3–5. Frame 3 will contain the picture from the ANI.B file.

64 Animator also lets you add sound effects in edit mode. Press D to add or clear the drum sound. Animator displays a small drum on the screen when the sound is present. To add musical notes in edit mode, press one of the number keys from 1–8 for low notes, and SHIFT plus 1–8 for high notes. A

Commodore 64 Accessories

Sale ☆ ☆ **SUPER** ☆ ☆ SAVE

Summer Sizzler Sale

Sale Positively Ends 9-10-85

Prices will go back to regular sale prices

	List	Reg. Sale Price	Summer Sizzler Sale
C128 Commodore Computer Expandable to 512K, runs C-64, CPM, and 7.0 Programs. (Add $10 Shipping)	$349.00	$299.00	$289.95*
15½" Commodore 150-170 CPS Printer Near Letter Quality, Multiple Pin Tractor / Friction Feed. Best Printer Value in U.S.A. (Add $17.50 Shipping)	$895.00	$299.00	$249.95*
Commodore-64 IEEE Interface Allows you to run Pet Peripherals on the C-64, including the One Megabyte Disk Drive and 15½" Printer	$109.95	$69.00	$65.95
Juki Printer/Typewriter Letter Quality, daisy wheel, use as typewriter and/or printer (auto correction) (Add $10 Shipping)	$349.00	$249.00	$229.95*
SCM 80 CPS Printer Tractor/Friction 10" Famous Name Printer does Graphics w/ Interface. (Add $10 Shipping)	$299.00	$159.00	$149.95*
Cardco G Plus Interface Converts Commodore to centronics for use with most printers, plus does Commodore graphics on graphic printers	$109.00	$59.00	$49.95
Alphacom 40 Column Printer Thermal technology - does graphics. (Add $10 Shipping) *Alphacom C-64 or Atori Interface $8.95.*	$99.00	$24.95	$22.95
190K Slim-Line Disk Drive Cooler, 20% faster, quieter than 1541 drive (Add $10 Shipping)	$249.00	$169.00	$139.95*
One Megabyte Disk Drive (1000K) Double sided drive hooks up to C-64 with IEEE interface, perfect as a second drive. (Add $10 Shipping)	$889.00	$199.00	$179.95*
13" Premium Quality Color Monitor Better than 1702, with separated video inputs. (Add $14.50 Shipping)	$399.00	$199.00	$169.95*
Voice Synthesizer Allows you to talk through your computer. Optional software lets you play talking adventure games (Zork, etc.)	$89.00	$49.00	$39.95
80 Column Board - 4 Slot Expander Allows you to program in 80 columns, plus gives you a 4 slot switch selectable expander	$149.00	$79.00	$59.95
Auto Dial Modem with Super Smart Software Upload, Download, Print, Catalog, This Package has it all!!! Best Modem Value in the U.S.A.!!	$129.95	$59.00	$39.95
Musicalc I Software The Best Musical Software for the C-64, allows you to change all parameters.	$59.95	$29.95	$14.95
Oil Barrons Software Better than Monopoly, comes with game board, disks and instruction manual. Strike Oil or Live in the Poor Form	$49.95	$19.95	$9.95

*** plus Software Coupon Discounts**

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!

VISA — MASTER CARD — C.O.D.

No C.O.D. to Canada, APO-FPO.

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

small note is displayed when a note is present. Clear a note by pressing 9. Control the duration of sounds from the main screen.

Press Q to exit 64 Animator. If you want to restart the program, you *must* load it again as described above. Don't reenter with SYS 49152.

Atari Version

"Atari Animator" is in two parts. First type in and save Program 3, then Program 4. If you're using cassette, substitute the following line for line 2 of program 3:

```
Ø6 2 PRINT "{DOWN}PRESS RETU
   RN":RUN "C:PART2"
```

Cassette users must also save Program 4 *on the same tape immediately after Program 3.* After both programs have been saved, plug in a joystick and read the general instructions above. When you're ready to continue, load and run Program 3 (be sure to rewind the tape to the beginning if you're using cassette). Program 3 loads the machine language portion of Animator, then automatically loads and runs Program 4.

The main screen provides 21 picture frames. Move the joystick up or down to select any of the values displayed at the bottom of the screen, and move it right or left to increase or decrease the selected value. Press E to go to the editing screen. In edit mode, move the blinking cursor around the drawing grid with the joystick. The fire button acts like a toggle: If you press it while the cursor is on a blank square, the cursor begins drawing; if you press it while the cursor is on a filled square, the cursor begins erasing.

To the right of the drawing grid is the current frame number and the picture in actual size. You can change to a different frame while in edit mode: Press the + key to increase the frame number and the − key to decrease it. Atari Animator also lets you copy a picture from one frame to another via the picture buffer. Press the G key to get (copy) the current picture into the buffer. Then change the frame number with the + or − key and press P to put (move) the picture from the buffer into the new frame.

Press the E key to exit the edit mode and return to the main screen. After setting the starting and ending frame numbers, press the fire button to begin the animation. To stop the animation, press the fire button again. While the cartoon is moving, you can change the figure's color by pressing any key (shifted keys provide additional colors).

The clear function (press C) clears the current frame in edit mode or all frames on the main screen. Atari Animator does not have automatic insert or delete functions, but you can accomplish the same thing with a series of individual get and put operations. You can save sequences on disk or tape, using any appropriate filename for disk.

Apple Version

"Apple Animator" runs on any Apple II-series computer with DOS 3.3 or ProDOS. The program requires two files on disk: the main BASIC program and a binary file (ANIMATOR2) that contains graphics data. Type in and save Programs 5 and 6, then run Program 5 to create ANIMATOR2 on disk. *You must run Program 5 before running Program 6 for the first time.* However, you don't need to run Program 5 each time you want use Program 6—only once to create the ANIMATOR2 file.

After running Program 5, read the general instructions above, then load and run Program 6. The program works much like the IBM PC/PCjr version, and most of its functions are self-prompting. Use the right and left arrow keys to pick the correct frame number before editing. For instance, if the frame number is 3 when you choose the edit function, frame 3 appears on the editing screen. To copy the picture in frame 3 into a different frame, enter the desired frame number when prompted.

When the editing screen appears, move the blinking cursor left, right, up, or down by pressing the J, L, I, and K keys, respectively. To put your drawing in the current frame, exit the edit mode with the save option (the quit option restores whatever that frame previously held). The remaining functions (load, save, insert, delete, invert, etc.) work as described in the general instructions above, except that Apple Animator uses no filename extenders for picture files.

TI Version

"TI Animator" is very similar to the IBM version. Be sure TI Extended BASIC is plugged in before typing and saving Program 7.

Since the TI-99/4A screen can display only 10 frames at once, the 20 frames are divided into two groups (1–10 and 11–20) on alternate screens. Press the N key to switch back and forth. You can change the color of the screen background or foreground drawing color by pressing the B or F keys.

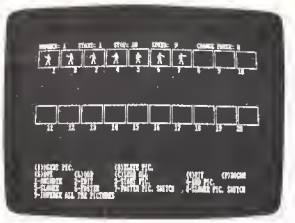
TI Animator can save animation files on disk or tape. When saving on tape, enter CS1 for the filename. When saving or loading from disk, be sure to type the prefix DSK1. at the beginning of each filename.

Program 1: Animator for IBM PC/PCjr

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
OP 5 DEF SEG=Ø:POKE 1Ø47,PEEK(1Ø
     47) OR 64
PK 3Ø KEY OFF:CLS:SCREEN 1:DEF S
     EG:POKE &H4E,1
CL 11Ø DEF SEG:POKE &H4E,2
OL 15Ø REM *** MAIN PROGRAM **
     *
AF 16Ø CLS:SCREEN 2:KEY OFF:FOR
     I=1 TO 1Ø:KEY I,"":NEXT
OJ 17Ø REM *** SET UP VARIABLES
     ***
GD 1BØ DIM A%(144),B%(144),O%(14
     4),E%(144),F%(144),C%(144
     ),G%(144),H%(144),I%(144)
     ,J%(144),K%(144),L%(144),
     M%(144),N%(144),O%(144),P
     %(144),Q%(144),R%(144),S%
     (144),T%(144),U%(144),A(2
     Ø,54):NUM=1:STA=1:EN=2Ø:S
     P=Ø:PL=1:GET(1,1Ø)-(54,3Ø
     ),U%
GL 19Ø X=9:Y=31:LOCATE 1,39:PRIN
     T"SPEED= ";SP;:LOCATE 1,1
     :PRINT"NUMBER=";NUM;:LOCA
     TE 1,14:PRINT"START=";STA
     ;:LOCATE 1,26:PRINT"STOP=
     ";EN;:LOCATE 1,55:PRINT"C
     HANGE PAUSE=";FAST
BE 2ØØ REM *** SET UP SCREEN *
     **
PB 21Ø LOCATE 22,1 :PRINT"1-ANIM
     ATE    2-EDIT        3-STA
     RT PIC.              4-END
     PIC.":LOCATE 23,1:PRINT"5
     -SLOWER     6-FASTER
     7-FASTER PIC. SWITCH    8
     -SLOWER PIC. SWITCH";:LOC
     ATE 24,1:PRINT"9-INVERSE
     ALL THE PICTURES";
FK 22Ø LOCATE 21,1:PRINT"(S)AVE
          (L)OAD          (C)LEA
     R ALL              (Q)UIT
          (P)ROGAM";:LOCATE 4,
     7Ø:FOR I=1 TO 1Ø:PRINT"
```

```
        ";I;:NEXT:LOCATE 13,79:
      FOR I=11 TO 20:PRINT"   "
      ;I;:NEXT
IH 230 LOCATE 20,1:PRINT"(I)NSER
      T PIC.          (O)ELE
      TE PIC.";
DK 240 FOR T=1 TO 2:FOR I=1 TO 5
      :LINE(A,X)-(A+55,Y),,B:LI
      NE(A+60,X)-(A+115,Y),,B:A
      =A+120:NEXT:A=0:X=80:Y=10
      2:NEXT
DD 250 REM *** READ PICTURES *
      **
DK 260 GET(1,10)-(54,30),AX:GET(
      61,10)-(114,30),BX:GET(12
      1,10)-(174,30),CX:GET(181
      ,10)-(234,30),DX:GET(241,
      10)-(294,30),EX:GET(301,1
      0)-(354,30),FX:GET(361,10
      )-(414,30),GX:GET(421,10)
      -(474,30),HX:GET(481,10)-
      (534,30),IX:GET(541,10)-(
      594,30),JX
EB 270 GET(1,81)-(54,101),KX:SET
      (61,81)-(114,101),LX:GET(
      121,81)-(174,101),MX:GET(
      181,81)-(234,101),NX:GET(
      241,81)-(294,101),OX:GET(
      301,81)-(354,101),PX:GET(
      361,81)-(414,101),QX:GET(
      421,81)-(474,101),RX:GET(
      481,81)-(534,101),SX:GET(
      541,81)-(594,101),TX
LK 280 REM *** WAIT FOR KEY **
      *
CI 290 A$=INKEY$:IF A$="" THEN 2
      90 ELSE IF LEN(A$)=2 THEN
       830 ELSE IF VAL(A$)>0 AN
      D VAL(A$)<10 THEN 430
OL 300 IF A$="P" THEN 3080
JD 310 IF A$<>"Q" THEN 350
PA 320 LOCATE 18,1:PRINT"ARE YOU
       SURE YOU WANT TO QUIT?
      (Y/N)"
JB 330 A$=INKEY$:IF A$="" THEN 3
      30 ELSE IF A$="Y" THEN SC
      REEN 0,0,0:CLS:END ELSE L
      OCATE 18,1:PRINT STRING$(
      50,32):GOTO 290
CD 340 REM *** CLEAR SCREEN **
      *
CC 350 IF A$<>"C" THEN 380
MM 360 LOCATE 17,1:PRINT"ARE YOU
       SURE  (Y/N)":DEF SEG:POK
      E 106,0
DD 370 A$=INKEY$:IF A$="" THEN 3
      70 ELSE IF A$="Y" THEN CL
      S:GOTO 190 ELSE LOCATE 17
      ,1:PRINT STRING$(66,32):G
      OTO 290
PH 380 IF A$="S" THEN 870 ELSE I
      F A$="L" THEN 990
OO 390 IF A$="O" THEN 2800 ELSE
      IF A$="I" THEN 2360
MM 400 GOTO 290
DF 410 IF FAST=0 THEN BEEP:GOTO
       290 ELSE FAST=FAST-1:LOCA
      TE 1,68:PRINT FAST;:GOTO
       290
JL 420 IF FAST=150 THEN BEEP: GO
      TO 290 ELSE FAST=FAST+1:L
      OCATE 1,68:PRINT FAST;:GO
      TO 290
KP 430 ON VAL(A$) GOTO 520,1080,
      470,500,480,450,410,420,3
      040
FH 440 REM *** SET SPEED ***
CI 450 IF SP=15 THEN BEEP ELSE S
      P=SP+1:LOCATE 1,46:PRINT
      SP
II 460 GOTO 290
DJ 470 STA=NUM:LOCATE 1,20:PRINT
       STA:GOTO 290
PP 480 IF SP=-15 THEN BEEP ELSE
```



"Animator for IBM PC/PCjr" takes advantage of extended BASIC graphics features such as GET and PUT.

```
      SP=SP-1:LOCATE 1,46:PRINT
       SP
IO 490 GOTO 290
HP 500 EN=NUM:LOCATE 1,31:PRINT
      EN:GOTO 290
LJ 510 REM *** ANIMATE PICTURES
       ***
BL 520 LOCATE 17,1:PRINT"PRESS S
      PACE BAR TO PAUSE MOVEMEN
      T":LOCATE 18,1:PRINT"USE
      ARROW KEYS FOR SPEED":IF
       STA>EN THEN QQ=-1 ELSE QQ
      =1
GB 530 FOR I=STA TO EN STEP QQ
CI 540 PL1=PL:IF PL+SP<1 THEN PL
      =570 ELSE IF PL+SP>580 TH
      EN PL=1
GM 550 PL=PL+SP
OK 560 LOCATE 1,8:PRINT I:ON I G
      OTO 630,640,650,660,670,6
      80,690,700,710,720,730,74
      0,750,760,770,780,790,800
      ,810,820
CN 570 FOR N=0 TO FAST*4:NEXT:A$
      =INKEY$:IF A$="" THEN 23
      40 ELSE IF A$<>"" AND LEN
      (A$)<>2 THEN LOCATE 1,8:P
      RINT NUM:LOCATE 17,1:PRIN
      T STRING$(150,32):GOTO 29
      0
DG 580 IF LEN(A$)<>2 THEN 620
FA 590 C=ASC(RIGHT$(A$,1)):IF C=
      77 THEN SP=SP+1 ELSE IF C
      =75 THEN SP=SP-1
EN 600 IF SP=-16 THEN SP=-15 ELS
      E IF SP=16 THEN SP=15
EL 610 LOCATE 1,46:PRINT SP
EJ 620 NEXT:GOTO 530
MD 630 PUT(PL1,52),UX,PSET:PUT(P
      L,52),AX:GOTO 570
DE 640 PUT(PL1,52),UX,PSET:PUT(P
      L,52),BX:GOTO 570
BK 650 PUT(PL1,52),UX,PSET:PUT(P
      L,52),CX:GOTO 570
DA 660 PUT(PL1,52),UX,PSET:PUT(P
      L,52),DX:GOTO 570
FG 670 PUT(PL1,52),UX,PSET:PUT(P
      L,52),EX:GOTO 570
IM 680 PUT(PL1,52),UX,PSET:PUT(P
      L,52),FX:GOTO 570
KC 690 PUT(PL1,52),UX,PSET:PUT(P
      L,52),GX:GOTO 570
LF 700 PUT(PL1,52),UX,PSET:PUT(P
      L,52),HX:GOTO 570
OL 710 PUT(PL1,52),UX,PSET:PUT(P
      L,52),IX:GOTO 570
AB 720 PUT(PL1,52),UX,PSET:PUT(P
      L,52),JX:GOTO 570
CH 730 PUT(PL1,52),UX,PSET:PUT(P
      L,52),KX:GOTO 570
FM 740 PUT(PL1,52),UX,PSET:PUT(P
      L,52),LX:GOTO 570
MD 750 PUT(PL1,52),UX,PBET:PUT(P
      L,52),MX:GOTO 570
KJ 760 PUT(PL1,52),UX,PSET:PUT(P
      L,52),NX:GOTO 570
MP 770 PUT(PL1,52),UX,PSET:PUT(P
      L,52),OX:GOTO 570
OF 780 PUT(PL1,52),UX,PSET:PUT(P
      L,52),PX:GOTO 570
BL 790 PUT(PL1,52),UX,PSET:PUT(P
      L,52),QX:GOTO 570
CO 800 PUT(PL1,52),UX,PSET:PUT(P
      L,52),RX:GOTO 570
EE 810 PUT(PL1,52),UX,PSET:PUT(P
      L,52),SX:GOTO 570
HK 820 PUT(PL1,52),UX,PSET:PUT(P
      L,52),TX:GOTO 570
JH 830 C=ASC(RIGHT$(A$,1)):IF C=
       77 THEN NUM=NUM+1 ELSE IF
       C=75 THEN NUM=NUM-1
FO 840 IF NUM=0 THEN NUM=20 ELSE
       IF NUM=21 THEN NUM=1
IM 850 LOCATE 1,8:PRINT NUM:GOTO
       290
PC 860 REM *** SAVE PICTURE **
      *
LO 870 CLS:ON ERROR GOTO 25010
HF 880 LOCATE 10,10:PRINT"F-FILE
      S     A-ABORT SAVE     AN
      Y OTHER KEY TO CONTINUE"
BH 890 A$=INKEY$:IF A$="" THEN 8
      90 ELSE IF A$="F" THEN FI
      LES"*.ANI" ELSE IF A$="A"
       THEN GOSUB 2310:GOTO 190
CJ 900 PRINT:PRINT:PRINT:INPUT"N
      AME OF FILE TO SAVE";A$:I
      F A$="" THEN GOSUB 2310:G
      OTO 190
JM 910 IF INSTR(A$,".")<>0 THEN
      CLS:LOCATE 9,10:PRINT"NO
      EXTENSION PLEASE.."GOTO
      880
II 920 IF LEN(A$)>8 THEN CLS:LOC
      ATE 9,10:PRINT"NO MORE TH
      AN 8 CHARACTERS PLEASE":G
      OTO 880
CD 930 IF VAL(RIGHT$(A$,1))>0 OR
       RIGHT$(A$,1)="0" THEN CL
      S:LOCATE 9,10:PRINT"THE F
      IRST CHARACTER CAN'T BE A
       NUMBER..":GOTO 880
BE 940 GOSUB 2310
BL 950 A$=A$+".ANI"
QI 960 DEF SEG=&H8800:BSAVE A$,0
      ,&H4000:PRINT"IT HAS BEEN
       SAVED.     PRESS ANY KEY
      TO CONTINUE":PRINT:PRINT
      :PRINT
QJ 970 A$=INKEY$:IF A$="" THEN 9
      70 ELSE CLS:GOSUB 2310:GO
      TO 190
AG 980 REM *** LOAD PICTURE **
      *
KJ 990 CLS:ON ERROR GOTO 25000
ID 1000 LOCATE 10,10:PRINT"F-FIL
      ES     A-ABORT LOAD
      ANY OTHER KEY TO CONTINU
      E"
CD 1010 A$=INKEY$:IF A$="" THEN
       1010 ELSE IF A$="F" THEN
       FILES"*.ANI" ELSE IF A$
      ="A" THEN GOSUB 2310:GOT
      O 190
PI 1020 PRINT:PRINT:PRINT:INPUT"
      NAME OF FILE TO LOAD";A$
      :IF A$="" THEN GOSUB 231
      0:GOTO 190
MI 1030 IF INSTR(A$,".")<>0 THEN
       CLS:LOCATE 9,10:PRINT"N
      O EXTENSION PLEASE..":GO
      TO 1000
BD 1040 IF LEN(A$)>8 THEN CLS:LO
      CATE 9,10:PRINT"NO MORE
      THAN 8 CHARACTERS PLEASE
      ":GOTO 1000
PK 1050 IF VAL(RIGHT$(A$,1))>0 O
      R RIGHT$(A$,1)="0" THEN
      CLS:LOCATE 9,10:PRINT"TH
```

```
   E FIRST CHARACTER CAN'T
   BE A NUMBER..":GOTO 1000
KE 1060 A$=A$+".ANI":DEF SEG=&H
   8000:BLOAD A$,0:ON ERROR
   GOTO 0:GOTO 190
LB 1070 REM ***  EDIT A PICTURE
    ***
JO 1080 LOCATE 16,1:PRINT"TYPE I
   N 21 TO ABORT":PRINT"NUM
   BER SET AT THE TOP OF TH
   E SCREEN IS PIC. TO READ
    FROM RETURN FOR SAME":L
   OCATE 18,1:INPUT"EDIT PI
   CTURE NUMBER";S:IF S<0 O
   R S>21 THEN BEEP:GOTO 10
   80
JC 1090 IF S=21 THEN CLS:GOSUB 2
   310:GOTO 190
IA 1100 IF S=0 THEN S=NUM
ED 1110 REM ***  PUT PICTURE TO
   EDIT ON SCREEN  ***
PB 1120 CLS:LOCATE 1,24:PRINT"WA
   IT...":ON NUM GOTO 1130,
   1140,1150,1160,1170,1180
   ,1190,1200,1210,1220,123
   0,1240,1250,1260,1270,12
   80,1290,1300,1310,1320
MG 1130 PUT(1,50),AX:GOTO 1340
PJ 1140 PUT(1,50),BX:GOTO 1340
ON 1150 PUT(1,50),CX:GOTO 1340
BP 1160 PUT(1,50),DX:GOTO 1340
CC 1170 PUT(1,50),EX:GOTO 1340
OF 1180 PUT(1,50),FX:GOTO 1340
FI 1190 PUT(1,50),GX:GOTO 1340
EP 1200 PUT(1,50),HX:GOTO 1340
FC 1210 PUT(1,50),IX:GOTO 1340
BF 1220 PUT(1,50),JX:GOTO 1340
II 1230 PUT(1,50),KX:GOTO 1340
JL 1240 PUT(1,50),LX:GOTO 1340
KD 1250 PUT(1,50),MX:GOTO 1340
LB 1260 PUT(1,50),NX:GOTO 1340
NE 1270 PUT(1,50),OX:GOTO 1340
NH 1280 PUT(1,50),PX:GOTO 1340
PK 1290 PUT(1,50),QX:GOTO 1340
OB 1300 PUT(1,50),RX:GOTO 1340
PE 1310 PUT(1,50),SX:GOTO 1340
AH 1320 PUT(1,50),TX:GOTO 1340
PD 1330 REM ***  GET ON-OFF POIN
   TS  ***
JD 1340 FOR I=1 TO 20:FOR X=1 TO
    54:A(I,X)=POINT(X,I+49)
PL 1350 NEXT:NEXT
KB 1360 REM ***  DRAW EDITING SC
   REEN  ***
IB 1370 FOR I=1 TO 20:LOCATE 3+I
   ,14:PRINT "..............
   ..............................
   ..............";:FOR
   J=1 TO 54:IF A(I,J)=1 TH
   EN LOCATE 3+I,13+J:PRINT
   "#"
PE 1380 NEXT:NEXT
CP 1390 LOCATE 1,30:PRINT"(Q)UIT
     (D)RAW   (M)OVE   (E)
   RASE   (C)LEAR   (S)AVE
     (I)NVERSE"
PL 1400 GOTO 1550
GF 1410 REM ***  PLACE CURSOR  *
   **
KO 1420 BLINK%=(BLINK%+1) MOD 20
   :IF BLINK%<10 THEN 1470
   ELSE 1440
PJ 1430 REM ***  CURSOR OFF  ***
GH 1440 IF A(ROW,COLUMN)=0 THEN
   CH$="." ELSE IF A(ROW,CO
   LUMN)=1 THEN CH$="#"
BF 1450 GOTO 1480
GN 1460 REM ***  CURSOR ON  ***
LO 1470 IF CURS=-1 THEN CH$="-"
   ELSE IF CURS=0 THEN CH$=
   "*" ELSE IF CURS=1 THEN
   CH$="+"
EI 1480 LOCATE 3+ROW,13+COLUMN:P
   RINT CH$;:RETURN
OC 1490 REM ***  REMOVE CURSOR
    ***
GN 1500 IF A(ROW,COLUMN)=0 THEN
   CH$="." ELSE IF A(ROW,CO
   LUMN)=1 THEN CH$="#"
CF 1510 LOCATE 3+ROW,13+COLUMN:P
   RINT CH$;:RETURN
DP 1520 LOCATE 24,18:PRINT "wait
   ";:FOR I=1 TO 20:LOCATE
   3+I,14:PRINT STRING$(54,
   46);
EN 1530 NEXT:ERASE A:DIM A(20,54
   ):LOCATE 24,18:PRINT"
   ";:PUT(1,50),UX,PSET:RE
   TURN
CO 1540 REM ***  SET CURSOR  ***
IJ 1550 ROW=1:COLUMN=1:CURS=0
NF 1560 REM ***  MAIN LOOP  ***
GH 1570 BLINK%=0:IF CURS=-1 THEN
    A(ROW,COLUMN)=0:PSET(CO
   LUMN,ROW+49),0 ELSE IF C
   URS=+1 THEN A(ROW,COLUMN
   )=1:PSET(COLUMN,ROW+49),
   1
IL 1580 GOSUB 1420
PB 1590 A$=INKEY$:DEF SEG:POKE 1
   06,0:IF LEN(A$)=0 THEN 1
   580 ELSE IF LEN(A$)=1 TH
   EN 1600 ELSE IF LEN(A$)=
   2 THEN 1720
OH 1600 CODE1=ASC(A$) AND &H5F
FC 1610 REM ***  READ KEYS  ***
OF 1620 IF CODE1=ASC("E") THEN 2
   040
PH 1630 IF CODE1=ASC("M") THEN 2
   050
GF 1640 IF CODE1=ASC("D") THEN 2
   060
JC 1650 IF CODE1=ASC("C") THEN 2
   0B0
OL 1660 IF CODE1=ASC("S") THEN 2
   100
GD 1670 IF CODE1=ASC("Q") THEN G
   OSUB 2310:GOTO 190
PP 1680 IF CODE1=ASC("I") THEN 1
   710
OB 1690 GOTO 1580
KG 1700 REM ***  INVERSE A PICTU
   RE  ***
LF 1710 GET(1,50)-(54,70),UX.PUT
   (1,50),UX,PRESET:GET(1,7
   5)-(54,95),UX:GOTO 1340
EL 1720 IF ASC(A$)<>0 THEN 1570
   ELSE CODE2=ASC(RIGHT$(A$
   ,1)):GOSUB 1500
HB 1730 REM ***  READ ARROW KEYS
    ***
IC 1740 IF CODE2=71 THEN 1840
PA 1750 IF CODE2=73 THEN 1870
MH 1760 IF CODE2=79 THEN 1900
KJ 1770 IF CODE2=81 THEN 1930
PJ 17B0 IF CODE2=72 THEN 1960
FN 1790 IF CODE2=75 THEN 1980
NL 1800 IF CODE2=77 THEN 2000
KP 1810 IF CODE2=80 THEN 2020
CA 1820 GOTO 1580
OK 1830 REM ***  MOVE THE CURSOR
    ***
EJ 1840 IF ROW=1 THEN ROW=21
FH 1850 IF COLUMN=1 THEN COLUMN=
   55
LE 1B60 ROW=ROW-1:COLUMN=COLUMN-
   1:GOTO 1570
EC 1870 IF ROW=1 THEN ROW=21
NN 1880 IF COLUMN=54 THEN COLUMN
   =0
ID 1890 ROW=ROW-1:COLUMN=COLUMN+
   1:GOTO 1570
LH 1900 IF ROW=20 THEN ROW=0
FH 1910 IF COLUMN=1 THEN COLUMN=
   55
JA 1920 ROW=ROW+1:COLUMN=COLUMN-
   1:GOTO 1570
NA 1930 IF ROW=20 THEN ROW=0
LO 1940 IF COLUMN=54 THEN COLUMN
   =0
GP 1950 ROW=ROW+1:COLUMN=COLUMN+
   1:GOTO 1570
EB 1960 IF ROW=1 THEN ROW=21
KN 1970 ROW=ROW-1:GOTO 1570
GC 1980 IF COLUMN=1 THEN COLUMN=
   55
ON 1990 COLUMN=COLUMN-1:GOTO 157
   0
JB 2000 IF COLUMN=54 THEN COLUMN
   =0
JD 2010 COLUMN=COLUMN+1:GOTO 157
   0
LN 2020 IF ROW=20 THEN ROW=0
GF 2030 ROW=ROW+1:GOTO 1570
OJ 2040 CURS=-1:GOTO 1570
KB 2050 CURS=0:GOTO 1570
CJ 2060 CURS=+1:GOTO 1570
NG 2070 REM ***  CLEAR THE PICTU
   RE  ***
BN 2080 GOSUB 1520:GOTO 1550
HE 2090 REM ***  SAVE THE PICTUR
   E  ***
PD 2100 LOCATE 1,24:PRINT"WAIT..
   .":ON B GOTO 2110,2120,2
   130,2140,2150,2160,2170,
   2180,2190,2200,2210,2220
   ,2230,2240,2250,2260,227
   0,22B0,2290,2300
JB 2110 GET(1,50)-(54,70),AX:GOS
   UB 2310:GOTO 190
LN 2120 GET(1,50)-(54,70),BX:GOS
   UB 2310:GOTO 190
NH 2130 GET(1,50)-(54,70),CX:GOS
   UB 2310:GOTO 190
OC 2140 GET(1,50)-(54,70),DX:GOS
   UB 2310:GOTO 190
ON 2150 GET(1,50)-(54,70),EX:GOS
   UB 2310:GOTO 190
CI 2160 GET(1,50)-(54,70),FX:GOS
   UB 2310:GOTO 190
OD 2170 GET(1,50)-(54,70),GX:GOS
   UB 2310:GOTO 190
FD 2180 GET(1,50)-(54,70),HX:GOS
   UB 2310:GOTO 190
HJ 2190 GET(1,50)-(54,70),IX:GOS
   UB 2310:GOTO 190
HI 2200 GET(1,50)-(54,70),JX:GOS
   UB 2310:GOTO 190
ID 2210 GET(1,50)-(54,70),KX:GOS
   UB 2310:GOTO 190
KO 2220 GET(1,50)-(54,70),LX:GOS
   UB 2310:GOTO 190
NJ 2230 GET(1,50)-(54,70),MX:GOS
   UB 2310:GOTO 190
NE 2240 GET(1,50)-(54,70),NX:GOS
   UB 2310:GOTO 190
PP 2250 GET(1,50)-(54,70),OX:GOS
   UB 2310:GOTO 190
BK 2260 GET(1,50)-(54,70),PX:GOS
   UB 2310:GOTO 190
CF 2270 GET(1,50)-(54,70),QX:GOS
   UB 2310:GOTO 190
EA 2280 GET(1,50)-(54,70),RX:GOS
   UB 2310:GOTO 190
GL 2290 GET(1,50)-(54,70),SX:GOS
   UB 2310:GOTO 190
BK 2300 GET(1,50)-(54,70),TX:GOS
   UB 2310:GOTO 190
GK 2310 CLS:PUT(1,10),AX:PUT(61,
   10),BX:PUT(121,10),CX:PU
   T(1B1,10),DX:PUT(241,10)
   ,EX:PUT(301,10),FX:PUT(3
   61,10),GX:PUT(421,10),HX
   :PUT(481,10),IX:PUT(541,
   10),JX
CA 2320 PUT(1,B1),KX:PUT(61,B1),
   LX:PUT(121,B1),MX:PUT(18
   1,B1),NX:PUT(241,B1),OX:
   PUT(301,B1),PX:PUT(361,B
   1),QX:PUT(421,B1),RX:PUT
   (481,B1),SX:PUT(541,B1),
   TX,PSET
```

```
JL 2330 RETURN
DM 2340 IF INKEY$="" THEN 2340 E
        LSE 620
PL 2350 REM ###  INSERT A BLANK
        PICTURE  ###
JH 2360 LOCATE 18,1:PRINT"ARE YO
        U SURE?  THIS WILL MOVE
        20 OFF THE END"
MC 2370 A$=INKEY$:IF A$="" THEN
        2370 ELSE IF A$<>"Y" THE
        N LOCATE 18,1:PRINT STRI
        NG$(50,32):GOTO 190
EF 2380 ON NUM GOTO 2390,2400,24
        10,2420,2430,2440,2450,2
        460,2470,2480,2490,2500,
        2510,2520,2530,2540,2550
        ,2560,2570,2820
MH 2390 GET(1,10)-(54,30),8%
LO 2400 GET(61,10)-(114,30),C%
FB 2410 GET(121,10)-(174,30),O%
IJ 2420 GET(181,10)-(234,30),E%
PN 2430 GET(241,10)-(294,30),F%
LM 2440 GET(301,10)-(354,30),G%
MP 2450 GET(361,10)-(414,30),H%
ED 2460 GET(421,10)-(474,30),I%
GG 2470 GET(481,10)-(534,30),J%
DK 2480 GET(541,10)-(594,30),K%
PO 2490 GET(1,81)-(54,101),L%
FA 2500 GET(61,81)-(114,101),M%
EK 2510 GET(121,81)-(174,101),N%
GD 2520 GET(181,81)-(234,101),O%
MO 2530 GET(241,81)-(294,101),P%
JD 2540 GET(301,81)-(354,101),Q%
LH 2550 GET(361,81)-(414,101),R%
DM 2560 GET(421,81)-(474,101),S%
FA 2570 GET(481,81)-(534,101),T%
AB 2580 CLS:ON NUM GOTO 2590,260
        0,2610,2620,2630,2640,26
        50,2660,2670,2680,2690,2
        700,2710,2720,2730,2740,
        2750,2760,2770,2780
LB 2590 GET(1,50)-(54,70),AX:GOS
        UB 2310:GOTO 190
LA 2600 GET(1,50)-(54,70),8X:GOS
        UB 2310:GOTO 190
ML 2610 GET(1,50)-(54,70),CX:GOS
        UB 2310:GOTO 190
DG 2620 GET(1,50)-(54,70),OX:GOS
        UB 2310:GOTO 190
AB 2630 GET(1,50)-(54,70),EX:GOS
        UB 2310:GOTO 190
CM 2640 GET(1,50)-(54,70),FX:GOS
        UB 2310:GOTO 190
DH 2650 GET(1,50)-(54,70),GX:GOS
        UB 2310:GOTO 190
FC 2660 GET(1,50)-(54,70),HX:GOS
        UB 2310:GOTO 190
HN 2670 GET(1,50)-(54,70),IX:GOS
        UB 2310:GOTO 190
JI 2680 GET(1,50)-(54,70),JX:GOS
        UB 2310:GOTO 190
KD 2690 GET(1,50)-(54,70),KX:GOS
        UB 2310:GOTO 190
KC 2700 GET(1,50)-(54,70),LX:GOS
        UB 2310:GOTO 190
MN 2710 GET(1,50)-(54,70),MX:GOS
        UB 2310:GOTO 190
DI 2720 GET(1,50)-(54,70),NX:GOS
        UB 2310:GOTO 190
PD 2730 GET(1,50)-(54,70),OX:GOS
        UB 2310:GOTO 190
BD 2740 GET(1,50)-(54,70),PX:GOS
        UB 2310:GOTO 190
DJ 2750 GET(1,50)-(54,70),QX:GOS
        UB 2310:GOTO 190
EE 2760 GET(1,50)-(54,70),RX:GOS
        UB 2310:GOTO 190
BP 2770 GET(1,50)-(54,70),SX:GOS
        UB 2310:GOTO 190
IK 2780 GET(1,50)-(54,70),TX:GOS
        UB 2310:GOTO 190
EH 2790 REM ###  DELETE A PICTUR
        E  ###
BM 2800 LOCATE 18,1:PRINT"ARE YO
        U SURE YOU WANT TO DELET
        E THIS NUMBER"
EF 2810 A$=INKEY$:IF A$="" THEN
        2810 ELSE IF A$<>"Y" THE
        N LOCATE 18,1:PRINT STRI
        NG$(50,32):GOTO 190
NH 2820 ON NUM GOTO 2830,2840,28
        50,2860,2870,2880,2890,2
        900,2910,2920,2930,2940,
        2950,2960,2970,2980,2990
        ,3000,3010,3020
JL 2830 GET(61,10)-(114,30),AX
DB 2840 GET(121,10)-(174,30),8%
FE 2850 GET(181,10)-(234,30),C%
NI 2860 GET(241,10)-(294,30),O%
IH 2870 GET(301,10)-(354,30),E%
LK 2880 GET(361,10)-(414,30),F%
CO 2890 GET(421,10)-(474,30),G%
CF 2900 GET(481,10)-(534,30),H%
KJ 2910 GET(541,10)-(594,30),I%
LB 2920 GET(1,81)-(54,101),J%
DL 2930 GET(61,81)-(114,101),K%
BD 2940 GET(121,81)-(174,101),L%
DH 2950 GET(181,81)-(234,101),M%
LM 2960 GET(241,81)-(294,101),N%
MM 2970 GET(301,81)-(354,101),O%
JA 2980 GET(361,81)-(414,101),P%
AF 2990 GET(421,81)-(474,101),Q%
GK 3000 GET(481,81)-(534,101),R%
HP 3010 GET(541,81)-(594,101),S%
IP 3020 CLS:GET(1,10)-(54,30),T%
        :GOSUB 2310:GOTO 190
PI 3030 REM ###  INVERSE ALL THE
         PICTURES  ###
IN 3040 CLS:PUT(1,10),AX,PRESET:
        PUT(61,10),8X,PRESET:PUT
        (121,10),CX,PRESET:PUT(1
        81,10),OX,PRESET:PUT(241
        ,10),EX,PRESET:PUT(301,1
        0),FX,PRESET:PUT(361,10)
        ,GX,PRESET:PUT(421,10),H
        X,PRESET:PUT(481,10),IX,
        PRESET:PUT(541,10),JX,PR
        ESET
MM 3050 PUT(1,81),KX,PRESET:PUT(
        61,81),LX,PRESET:PUT(121
        ,81),MX,PRESET:PUT(181,8
        1),NX,PRESET:PUT(241,81)
        ,OX,PRESET:PUT(301,81),P
        X,PRESET:PUT(361,81),QX,
        PRESET:PUT(421,81),RX,PR
        ESET:PUT(481,81),SX,PRES
        ET:PUT(541,81),TX,PRESET
EG 3060 GOTO 190
IF 3070 REM ###  MAKE A PROGRAM
        ###
BN 3080 LOCATE 18,1:PRINT"ARE YO
        U SURE YOU WANT TO MAKE
        THIS SET-UP INTO A PROGR
        AM?"
PJ 3090 A$=INKEY$:IF A$="" THEN
        3090 ELSE IF A$="Y" THEN
         3100 ELSE LOCATE 18,1:P
        RINT STRING$(66,32):GOTO
        290
MN 3100 G=30:OPEN "O",#1,"PRG.8A
        S"
PL 3110 PRINT #1,"10 CLS:KEY OFF
        :SCREEN 2:SP="+STR$(SP)+
        ":PL=1"
JL 3120 IF STA>EN THEN QQ=-1 ELS
        E QQ=1
EP 3130 A$="20 OIM ":FOR I=STA T
        O EN STEP QQ:IF I<>STA T
        HEN A$=A$+","
KE 3140 ON I GOTO 3150,3160,3170
        ,3180,3190,3200,3210,322
        0,3230,3240,3250,3260,32
        70,3280,3290,3300,3310,3
        320,3330,3340
BM 3150 A$=A$+"AX(144)":GOTO 335
        0
CM 3160 A$=A$+"8X(144)":GOTO 335
        0
DM 3170 A$=A$+"CX(144)":GOTO 335
        0
EM 3180 A$=A$+"OX(144)":GOTO 335
        0
FM 3190 A$=A$+"EX(144)":GOTO 335
        0
EA 3200 A$=A$+"FX(144)":GOTO 335
        0
FA 3210 A$=A$+"GX(144)":GOTO 335
        0
GA 3220 A$=A$+"HX(144)":GOTO 335
        0
HA 3230 A$=A$+"IX(144)":GOTO 335
        0
IA 3240 A$=A$+"JX(144)":GOTO 335
        0
JA 3250 A$=A$+"KX(144)":GOTO 335
        0
KA 3260 A$=A$+"LX(144)":GOTO 335
        0
LA 3270 A$=A$+"MX(144)":GOTO 335
        0
MA 3280 A$=A$+"NX(144)":GOTO 335
        0
NA 3290 A$=A$+"OX(144)":GOTO 335
        0
ME 3300 A$=A$+"PX(144)":GOTO 335
        0
NE 3310 A$=A$+"QX(144)":GOTO 335
        0
DE 3320 A$=A$+"RX(144)":GOTO 335
        0
PE 3330 A$=A$+"SX(144)":GOTO 335
        0
AE 3340 A$=A$+"TX(144)":GOTO 335
        0
AD 3350 NEXT
NI 3360 A$=A$+",UX(144)":PRINT #
        1,A$:PRINT #1,"30 GET(1,
        1)-(54,20),UX"
KD 3370 IF STA>EN THEN QQ=-1 ELS
        E QQ=1
KG 3380 FOR P=STA TO EN STEP QQ
CE 3390 CLS:ON P GOTO 3400,3410,
        3420,3430,3440,3450,3460
        ,3470,3480,3490,3500,351
        0,3520,3530,3540,3550,35
        60,3570,3580,3590
PI 3400 PUT(1,50),AX:GOTO 3600
QL 3410 PUT(1,50),8X:GOTO 3600
BD 3420 PUT(1,50),CX:GOTO 3600
CB 3430 PUT(1,50),OX:GOTO 3600
DE 3440 PUT(1,50),EX:GOTO 3600
EH 3450 PUT(1,50),FX:GOTO 3600
GK 3460 PUT(1,50),GX:GOTO 3600
HN 3470 PUT(1,50),HX:GOTO 3600
IA 3480 PUT(1,50),IX:GOTO 3600
JD 3490 PUT(1,50),JX:GOTO 3600
JK 3500 PUT(1,50),KX:GOTO 3600
KN 3510 PUT(1,50),LX:GOTO 3600
LA 3520 PUT(1,50),MX:GOTO 3600
MD 3530 PUT(1,50),NX:GOTO 3600
NG 3540 PUT(1,50),OX:GOTO 3600
PJ 3550 PUT(1,50),PX:GOTO 3600
QM 3560 PUT(1,50),QX:GOTO 3600
RP 3570 PUT(1,50),RX:GOTO 3600
CC 3580 PUT(1,50),SX:GOTO 3600
SF 3590 PUT(1,50),TX:GOTO 3600
NN 3600 FOR X=1 TO 54:G=G+10:A$=
        STR$(G):A$=RIGHT$(A$,LEN
        (A$)-1):FOR I=1 TO 20:IF
         POINT(X,I+49)=1 THEN A$
        =A$+"PSET(":8$=STR$(X):
        A$=A$+RIGHT$(8$,LEN(8$)-
        1):A$=A$+",":8$=STR$(I+4
        9):A$=A$+RIGHT$(8$,LEN(8
        $)-1):A$=A$+")"
FH 3610 NEXT:IF LEN(A$)>6 THEN 8
        $=LEFT$(A$,LEN(STR$(G))-
        1):A$=RIGHT$(A$,(LEN(A$)
        -LEN(8$))-1):A$=8$+" "+A
        $:PRINT #1,A$ ELSE G=G-1
        0
AA 3620 NEXT
```

```
BD 3630 G=G+10:A$=STR$(G):A$=RIG
        HT$(A$,LEN(A$)-1):ON P G
        OTO 3640,3650,3660,3670,
        3680,3690,3700,3710,3720
        ,3730,3740,3750,3760,377
        0,3780,3790,3800,3810,38
        20,3830
GM 3640 A$=A$+" GET(1,50)-(54,70
        ),AX":GOTO 3840
CP 3650 A$=A$+" GET(1,50)-(54,70
        ),BX":GOTO 3840
EC 3660 A$=A$+" GET(1,50)-(54,70
        ),CX":GOTO 3840
GF 3670 A$=A$+" GET(1,50)-(54,70
        ),DX":GOTO 3840
JI 3680 A$=A$+" GET(1,50)-(54,70
        ),EX":GOTO 3840
LL 3690 A$=A$+" GET(1,50)-(54,70
        ),FX":GOTO 3840
LC 3700 A$=A$+" GET(1,50)-(54,70
        ),GX":GOTO 3840
NF 3710 A$=A$+" GET(1,50)-(54,70
        ),HX":GOTO 3840
QI 3720 A$=A$+" GET(1,50)-(54,70
        ),IX":GOTO 3840
CL 3730 A$=A$+" GET(1,50)-(54,70
        ),JX":GOTO 3840
ED 3740 A$=A$+" GET(1,50)-(54,70
        ),KX":GOTO 3840
BB 3750 A$=A$+" GET(1,50)-(54,70
        ),LX":GOTO 3840
IE 3760 A$=A$+" GET(1,50)-(54,70
        ),MX":GOTO 3840
KH 3770 A$=A$+" GET(1,50)-(54,70
        ),NX":GOTO 3840
MK 3780 A$=A$+" GET(1,50)-(54,70
        ),OX":GOTO 3840
PN 3790 A$=A$+" GET(1,50)-(54,70
        ),PX":GOTO 3840
PE 3800 A$=A$+" GET(1,50)-(54,70
        ),QX":GOTO 3840
BH 3810 A$=A$+" GET(1,50)-(54,70
        ),RX":GOTO 3840
EK 3820 A$=A$+" GET(1,50)-(54,70
        ),SX":GOTO 3840
GN 3830 A$=A$+" GET(1,50)-(54,70
        ),TX":GOTO 3840
PN 3840 A$=A$+":CLS":PRINT #1,A$
        :NEXT
KG 3850 IF STA=EN THEN STN=1:EA=
        1:GOTO 3880
MK 3860 IF STA>EN THEN STN=STA-E
        N:EA=1 ELSE EA=EN-STA:ST
        N=1
ED 3870 IF EA+QQ=0 THEN EA=EA-QQ
OF 3880 G=G+10:R=G:A$=RIGHT$(STR
        $(G),LEN(STR$(G))-1)+" F
        OR I="+RIGHT$(STR$(STN),
        LEN(STR$(STN))-1)+" TO"+
        STR$(EA+QQ)+" STEP"+STR$
        (QQ):PRINT #1,A$:G=G+10
GM 3890 A$=RIGHT$(STR$(G),LEN(ST
        R$(G))-1)+" FOR P=1 TO";
        A$=A$+STR$(INT(FAST*4.2)
        )+":NEXT:PL1=PL:IF PL+SP
        <1 THEN PL=570 ELSE IF P
        L+SP>580 THEN PL=1"
MJ 3900 G=G+10:PRINT #1,A$
FG 3910 A$=RIGHT$(STR$(G),LEN(ST
        R$(G))-1)+" PL=PL+SP":PR
        INT #1,A$:G=G+10
CK 3920 A$=RIGHT$(STR$(G),LEN(ST
        R$(G))-1)+" ON I GOTO":X
        =G+10:A$=A$+STR$(X):FOR
        I=STA TO EN+(QQ*-1) STEP
         QQ:X=X+10:A$=A$+","+RIG
        HT$(STR$(X),LEN(STR$(X))
        -1):NEXT:PRINT#1,A$
KL 3930 IF STA>EN THEN Q=STA:W=E
        N ELSE IF EN>STA THEN Q=
        EN:W=STA
DL 3940 FOR P=W TO Q
PP 3950 G=G+10:A$=RIGHT$(STR$(G)
        ,LEN(STR$(G))-1):ON P GO
        TO 3960,3970,3980,3990,4
        000,4010,4020,4030,4040,
        4050,4060,4070,4080,4090
        ,4100,4110,4120,4130,414
        0,4150
IE 3960 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),AX%":GOTO
         4160
LE 3970 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),BX%":GOTO
         4160
OE 3980 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),CX%":GOTO
         4160
BE 3990 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),DX%":GOTO
         4160
BF 4000 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),EX%":GOTO
         4160
EF 4010 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),FX%":GOTO
         4160
HF 4020 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),GX%":GOTO
         4160
KF 4030 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),HX%":GOTO
         4160
MF 4040 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),IX%":GOTO
         4160
AF 4050 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),JX%":GOTO
         4160
DF 4060 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),KX%":GOTO
         4160
GF 4070 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),LX%":GOTO
         4160
JF 4080 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),MX%":GOTO
         4160
MF 4090 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),NX%":GOTO
         4160
OJ 4100 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),OX%":GOTO
         4160
BJ 4110 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),PX%":GOTO
         4160
EJ 4120 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),QX%":GOTO
         4160
HJ 4130 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),RX%":GOTO
         4160
KJ 4140 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),SX%":GOTO
         4160
MJ 4150 A$=A$+" PUT(PL1,52),U%,P
        SET:PUT(PL,52),TX%":GOTO
         4160
DO 4160 A$=A$+"GOTO"+STR$(X+10):
        PRINT #1,A$:NEXT:G=X+10
LM 4170 A$=RIGHT$(STR$(G),LEN(ST
        R$(G))-1):A$=A$+" NEXT:G
        OTO"+STR$(R):PRINT#1,A$
NA 4180 CLOSE #1:CLS:PRINT"BEFOR
        E YOU DO ANYTHING ELSE L
        OAD THE PROGRAM <PRG> AN
        D THEN SAVE IT UNDER THE
           NAME YOU WANT":END
GD 25000 GOSUB 25020:RESUME 990
BH 25010 GOSUB 25020:RESUME 870
PJ 25020 PRINT:PRINT "
                              OIS
        K ERROR #"ERR
QP 25030 PRINT:PRINT "
                          HIT SPACE
        BAR TO CONTINUE"
CM 25040 A$=INKEY$:IF
        EN 25040 ELSE :
```

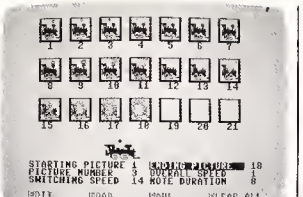
Program 2: 64 Animator

Version by Kevin Mykytyn, Editorial Programmer

Please refer to the "MLX" article in this issue before entering the following listing.

```
49152 :032,238,203,032,142,203,082
49158 :160,127,185,123,206,153,192
49164 :064,003,136,016,247,076,042
49170 :046,192,000,001,001,015,017
49176 :001,012,008,000,000,000,045
49182 :000,000,000,000,000,000,030
49188 :006,000,000,000,000,000,042
49194 :000,000,000,000,032,077,151
49200 :192,169,000,141,028,192,002
49206 :032,070,193,032,044,193,106
49212 :032,070,194,032,215,193,028
49218 :032,033,197,162,255,032,009
49224 :205,194,076,060,192,120,151
49230 :169,127,141,013,220,169,149
49236 :105,141,020,003,169,192,202
49242 :141,021,003,169,129,141,182
49248 :026,208,169,027,141,017,172
49254 :208,088,096,169,001,141,037
49260 :025,208,173,044,192,208,190
49266 :069,173,018,208,201,010,025
49272 :176,011,162,225,160,059,145
49278 :169,083,141,018,208,208,185
49284 :029,201,085,176,011,162,028
49290 :232,160,099,169,130,141,045
49296 :018,208,208,014,162,239,225
49302 :160,139,169,000,141,018,009
49308 :208,240,003,140,018,208,205
49314 :152,160,012,153,001,208,080
49320 :136,136,016,249,160,000,097
49326 :138,153,248,007,200,232,128
49332 :192,007,208,246,173,013,251
49338 :220,041,001,240,081,164,165
49344 :203,204,043,192,240,056,106
49350 :140,043,192,185,129,235,098
49356 :201,088,208,008,173,029,143
49362 :208,073,128,141,029,208,229
49368 :201,089,208,008,173,023,150
49374 :208,073,128,141,023,208,235
49380 :201,049,144,022,201,057,134
49386 :176,018,056,174,141,002,033
49392 :224,004,208,010,233,049,200
49398 :160,007,153,039,208,136,181
49404 :016,250,201,133,208,011,047
49410 :173,045,192,240,006,032,178
49416 :019,193,032,238,203,076,001
49422 :049,234,076,188,254,173,220
49428 :136,002,073,012,141,136,008
49434 :002,173,024,208,073,048,042
49440 :141,024,208,173,021,208,039
49446 :073,127,141,021,208,096,192
49452 :169,127,141,021,208,169,111
49458 :064,141,016,208,160,006,133
49464 :162,012,185,047,204,157,055
49470 :000,208,202,202,136,016,058
49476 :245,096,169,147,032,210,199
49482 :255,169,015,141,033,208,127
49488 :162,001,160,002,024,032,205
49494 :240,255,032,197,193,162,141
49500 :006,160,002,024,032,240,044
49506 :255,032,197,193,162,011,180
49512 :160,002,024,032,240,255,049
49518 :032,197,193,162,004,160,090
49524 :002,024,032,240,255,169,070
49530 :085,160,204,032,030,171,036
49536 :162,009,160,002,024,032,005
49542 :240,255,169,120,160,204,002
49548 :032,030,171,162,014,160,197
49554 :002,024,032,240,255,169,100
49560 :155,160,204,032,030,171,136
49566 :032,165,193,032,181,193,186
49572 :096,162,019,160,001,024,114
49578 :032,240,255,169,190,160,192
49584 :204,032,030,171,096,162,103
49590 :022,160,001,024,032,240,149
49596 :255,169,116,160,205,032,101
49602 :030,171,096,169,007,141,040
49608 :020,192,169,054,160,204,231
49614 :032,030,171,206,020,192,089
49620 :208,244,096,162,019,160,077
49626 :018,024,032,240,255,169,188
```

```
49632 :000,174,021,192,032,061,192
49638 :194,162,019,160,038,024,059
49644 :032,240,255,169,000,174,082
49650 :024,192,032,061,194,162,139
49656 :020,160,018,024,032,240,230
49662 :255,169,000,174,022,192,042
49668 :032,061,194,032,164,196,171
49674 :162,021,160,018,024,032,171
49680 :240,255,169,000,174,023,109
49686 :192,202,032,061,194,162,097
49692 :021,160,038,024,032,240,031
49698 :255,169,000,174,026,192,082
49704 :224,016,144,002,162,015,091
49710 :142,038,192,032,061,194,193
49716 :173,038,192,009,016,141,109
49722 :012,212,096,032,205,189,036
49728 :169,032,032,210,255,096,090
49734 :174,028,192,173,000,220,089
49740 :074,176,014,032,159,194,213
49746 :202,016,002,162,006,032,246
49752 :159,194,076,154,194,074,171
49758 :176,016,032,159,194,232,135
49764 :224,007,144,002,162,001,128
49770 :032,159,194,076,154,194,147
49776 :074,176,013,189,020,192,008
49782 :201,001,240,003,222,028,037
49788 :192,076,154,194,074,176,222
49794 :013,189,020,192,201,021,254
49800 :240,003,254,020,192,076,153
49806 :154,194,074,176,007,138,117
49812 :072,032,225,194,104,170,177
49818 :142,028,192,096,096,072,012
49824 :138,072,010,170,189,051,022
49830 :205,133,251,189,052,205,177
49836 :133,252,160,015,177,251,136
49842 :073,128,145,251,136,016,159
49848 :247,104,170,104,096,169,050
49854 :001,141,028,192,169,000,209
49860 :160,004,153,021,192,136,094
49866 :016,250,096,160,100,136,192
49872 :208,253,202,208,248,096,143
49878 :032,205,194,173,000,220,014
49884 :041,016,240,246,096,032,123
49890 :214,194,169,255,141,045,220
49896 :192,141,021,208,169,000,195
49902 :141,029,192,169,180,141,066
49908 :015,208,173,036,192,141,241
49914 :046,208,173,024,192,205,074
49920 :021,192,176,003,238,029,147
49926 :192,173,021,192,141,022,235
49932 :192,169,000,141,037,192,231
49938 :169,021,056,237,023,192,204
49944 :141,032,192,201,021,208,051
49950 :003,238,037,192,173,025,186
49956 :192,141,034,192,173,032,032
49962 :192,240,020,173,037,192,128
49968 :208,113,206,033,192,208,240
49974 :108,169,100,141,033,192,029
49980 :206,032,192,208,098,169,197
49986 :021,056,237,023,192,141,224
49992 :032,192,173,021,192,205,119
49998 :024,192,240,041,173,022,002
50004 :192,205,024,192,208,008,145
50010 :173,021,192,141,022,192,063
50016 :208,025,173,029,192,240,195
50022 :011,173,022,192,056,233,021
50028 :001,141,022,192,208,009,169
50034 :173,022,192,024,105,001,119
50040 :141,022,192,168,185,220,024
50046 :055,240,010,169,128,141,101
50052 :004,212,169,129,141,004,023
50058 :212,185,250,055,240,019,075
50064 :141,008,212,185,024,056,002
50070 :141,007,212,169,032,141,084
50076 :011,212,169,033,141,011,221
50082 :212,169,000,141,030,192,138
50088 :173,034,192,240,005,206,250
50094 :034,192,208,060,173,025,098
50100 :192,056,233,011,016,008,184
50106 :238,030,192,073,255,024,230
50112 :105,001,141,034,192,173,070
50118 :034,192,170,240,007,024,097
50124 :109,034,192,202,208,249,174
50130 :141,035,192,173,030,192,205
50136 :240,058,173,031,192,056,198
50142 :237,035,192,141,031,192,026
50148 :173,014,208,233,000,141,229
50154 :014,208,201,255,208,090,186
50160 :173,016,208,041,128,208,246
50166 :015,169,085,141,014,208,110
50172 :173,016,208,009,128,141,159
```



"64 Animator" uses raster interrupt techniques to display up to 22 sprites at once.

```
50178 :016,208,208,068,173,016,179
50184 :208,041,127,141,016,208,237
50190 :206,014,208,076,074,196,020
50196 :173,031,192,024,109,035,072
50202 :192,141,031,192,173,014,001
50208 :208,105,000,141,014,208,196
50214 :208,010,173,016,208,009,150
50220 :128,141,016,208,208,024,001
50226 :201,085,208,020,173,016,241
50232 :208,041,128,240,013,169,087
50238 :010,141,014,208,173,016,112
50244 :208,041,127,141,016,208,041
50250 :234,173,000,220,074,074,081
50256 :074,176,016,173,025,192,224
50262 :240,054,206,027,192,208,245
50268 :049,206,025,192,076,142,014
50274 :196,074,176,018,173,025,248
50280 :192,201,021,240,033,238,005
50286 :027,192,208,028,238,025,060
50292 :192,076,142,196,074,176,204
50298 :019,169,000,141,045,192,176
50304 :173,136,002,201,004,240,116
50310 :003,032,019,193,032,214,115
50316 :194,096,032,164,196,032,086
50322 :235,196,173,022,192,168,108
50328 :024,105,224,141,255,007,140
50334 :141,255,011,076,040,195,108
50340 :169,000,141,069,219,141,135
50346 :070,219,141,071,219,173,039
50352 :025,192,056,233,011,072,253
50358 :016,014,169,045,141,069,124
50364 :007,104,073,255,024,105,244
50370 :001,076,204,196,169,032,104
50376 :141,069,007,104,201,010,220
50382 :144,016,072,169,049,141,029
50388 :070,007,104,056,233,010,180
50394 :009,048,141,071,007,096,078
50400 :009,048,141,070,007,169,156
50406 :032,141,071,007,096,173,238
50412 :022,192,201,020,144,016,063
50418 :072,169,050,141,050,007,219
50424 :104,056,233,020,009,048,206
50430 :141,051,007,096,201,010,248
50436 :144,016,072,169,049,141,083
50442 :050,007,104,056,233,010,214
50448 :009,048,141,051,007,096,112
50454 :009,048,141,050,007,169,190
50460 :032,141,051,007,096,032,131
50466 :228,255,201,069,208,003,230
50472 :076,184,199,201,067,208,207
50478 :038,032,002,204,176,033,019
50484 :169,064,133,251,169,056,126
50490 :133,252,032,126,203,198,234
50496 :252,032,126,203,160,021,090
50502 :169,000,153,220,055,153,052
50508 :250,055,153,024,056,136,238
50514 :016,244,096,201,073,208,152
50520 :096,169,128,133,251,169,010
50526 :061,133,252,169,064,133,138
50532 :253,169,061,133,254,169,115
50538 :022,056,237,022,192,170,037
50544 :240,071,160,063,177,253,052
50550 :145,251,136,016,249,165,056
50556 :251,056,233,064,133,251,088
50562 :165,252,233,000,133,252,141
50568 :165,253,056,233,064,133,016
50574 :253,165,254,233,000,133,156
50580 :254,202,208,218,162,020,188
50586 :189,250,055,072,189,024,165
50592 :056,072,189,220,055,232,216
```

```
50598 :157,220,055,104,157,024,115
50604 :056,104,157,250,055,202,228
50610 :202,236,022,192,176,226,208
50616 :096,201,068,208,100,173,006
50622 :022,192,032,165,199,165,197
50628 :251,024,105,064,133,253,002
50634 :165,252,105,000,133,254,087
50640 :169,022,056,237,022,192,138
50646 :170,240,072,160,063,177,072
50652 :253,145,251,136,016,249,246
50658 :165,251,024,105,064,133,200
50664 :251,165,252,105,000,133,114
50670 :252,165,253,024,105,064,077
50676 :133,253,165,254,105,000,130
50682 :133,254,202,016,218,174,223
50688 :022,192,232,189,250,055,172
50694 :072,189,024,056,072,189,096
50700 :220,055,202,157,220,055,153
50706 :104,157,024,056,104,157,108
50712 :250,055,232,232,224,021,014
50718 :144,227,096,201,083,208,221
50724 :044,032,146,199,032,063,040
50730 :199,169,220,133,253,169,161
50736 :055,133,254,173,024,192,111
50742 :032,165,199,165,251,024,122
50748 :105,064,170,165,252,105,153
50754 :000,168,169,253,032,216,136
50760 :255,032,233,198,104,104,230
50766 :076,046,192,201,076,208,109
50772 :019,032,146,199,032,063,063
50778 :199,169,000,032,213,255,190
50784 :032,233,198,104,104,076,075
50790 :046,192,201,036,208,099,116
50796 :032,146,199,169,001,162,049
50802 :008,160,000,032,186,255,243
50808 :169,008,162,059,160,206,116
50814 :032,189,255,032,192,255,057
50820 :032,233,198,162,001,032,022
50826 :198,255,032,207,255,032,093
50832 :207,255,169,013,032,210,006
50838 :255,032,207,255,032,207,114
50844 :255,032,233,198,176,032,058
50850 :032,207,255,170,032,207,041
50856 :255,032,205,189,169,032,026
50862 :032,210,255,032,207,255,141
50868 :240,220,032,210,255,165,022
50874 :203,201,064,208,250,076,164
50880 :177,198,169,001,032,195,196
50886 :255,032,204,255,104,104,128
50892 :076,046,192,201,081,208,240
50898 :006,032,002,204,176,001,119
50904 :000,096,169,000,160,206,079
50910 :032,030,171,032,228,255,202
50916 :201,013,208,249,096,141,112
50922 :038,192,024,165,144,240,013
50928 :074,041,128,240,020,169,144
50934 :099,160,206,032,030,171,176
50940 :169,001,032,195,255,032,168
50946 :218,198,104,104,076,202,136
50952 :198,169,015,162,008,168,216
50958 :032,186,255,169,000,032,176
50964 :189,255,032,192,255,162,081
50970 :015,032,198,255,169,013,196
50976 :032,210,255,032,207,255,255
50982 :072,032,210,255,104,201,144
50988 :013,208,244,169,015,032,213
50994 :195,255,032,204,255,032,255
51000 :218,198,056,173,038,192,163
51006 :096,169,210,160,205,032,166
51012 :030,171,032,249,171,173,126
51018 :000,002,208,005,104,104,241
51024 :076,076,198,160,255,200,021
51030 :185,000,002,153,038,002,210
51036 :208,247,162,005,189,060,195
51042 :206,157,032,002,200,202,129
51048 :016,246,152,162,032,160,104
51054 :002,032,189,255,169,229,218
51060 :160,205,032,030,171,032,234
51066 :228,255,201,084,208,004,078
51072 :162,001,208,006,201,068,006
51078 :208,241,162,008,169,002,156
51084 :160,001,032,186,255,096,102
51090 :032,091,255,032,138,255,181
51096 :169,024,141,024,208,169,119
51102 :251,160,205,032,030,171,239
51108 :096,024,105,224,133,251,229
51114 :169,000,133,252,162,006,124
51120 :006,251,038,252,202,208,109
51126 :249,096,169,001,141,044,114
51132 :192,169,055,141,011,208,196
51138 :141,013,208,169,035,141,133
```

```
51144 :012,208,169,255,141,010,227
51150 :208,169,224,141,021,208,153
51156 :169,195,141,016,208,169,086
51162 :020,141,014,208,169,085,087
51168 :141,015,208,169,084,133,206
51174 :247,169,004,133,248,169,176
51180 :020,141,000,200,141,002,236

51186 :208,169,165,141,001,208,110
51192 :169,190,141,003,208,169,104
51198 :013,141,248,007,169,014,078
51204 :141,249,007,032,170,202,037
51210 :172,022,192,185,220,055,088
51216 :240,020,173,021,208,009,175
51222 :001,141,021,208,169,128,178
51228 :141,004,212,169,129,141,056
51234 :004,212,208,008,173,021,148
51240 :208,041,254,141,021,208,145
51246 :172,022,192,185,250,055,154
51252 :240,030,141,008,212,185,100
51258 :024,056,141,007,212,169,155
51264 :032,141,011,212,169,033,150
51270 :141,011,212,173,021,208,068
51276 :009,002,141,021,208,076,021
51282 :092,200,173,021,208,041,049
51288 :253,141,021,208,032,194,169
51294 :202,032,118,203,169,084,134
51300 :133,253,133,141,169,216,121
51306 :133,254,133,142,173,022,195
51312 :192,024,105,224,141,255,029
51318 :007,056,233,001,141,253,041
51324 :007,024,105,002,141,254,145
51330 :007,173,022,192,032,165,209
51336 :199,165,251,133,139,165,164
51342 :252,133,140,169,000,141,209
51348 :038,192,169,003,141,039,218
51354 :192,172,038,192,177,251,152
51360 :140,038,192,160,000,162,084
51366 :008,010,144,006,072,169,063
51372 :001,145,253,104,230,253,134
51378 :208,002,230,254,202,208,002
51384 :238,238,038,192,173,038,077
51390 :192,201,064,240,021,206,090
51396 :039,192,208,211,165,253,240
51402 :024,105,016,133,253,165,130
51408 :254,105,000,133,254,076,006
51414 :150,200,165,247,133,249,078
51420 :165,248,133,250,160,000,152
51426 :177,247,073,128,145,247,219
51432 :162,128,032,205,194,073,002
51438 :128,145,247,173,000,220,127
51444 :074,176,015,072,165,247,225
51450 :056,233,040,133,247,165,100
51456 :248,233,000,133,248,104,198
51462 :074,176,015,072,165,247,243
51468 :024,105,040,133,247,165,244
51474 :248,105,000,133,248,104,088
51480 :074,176,015,072,165,247,005
51486 :056,233,001,133,247,165,097
51492 :248,233,000,133,248,104,234
51498 :074,176,015,072,165,247,023
51504 :024,105,001,133,247,165,211
51510 :248,105,000,133,248,104,124
51516 :074,176,038,165,249,024,018
51522 :105,000,133,078,165,250,029
51528 :105,212,133,073,179,041,047
51534 :192,208,012,169,001,141,033

51540 :041,192,177,078,073,001,134
51546 :141,040,192,173,040,192,100
51552 :145,078,076,106,201,169,103
51558 :000,141,041,192,177,247,132
51564 :201,032,208,008,165,249,203
51570 :133,247,165,250,133,248,010
51576 :032,228,255,201,082,208,102
51582 :013,169,000,141,044,192,173
51588 :169,129,141,026,208,076,113
51594 :049,192,201,147,208,003,170
51600 :032,170,202,201,043,208,232
51606 :013,173,022,192,201,021,004
51612 :240,006,238,022,192,076,162
51618 :010,200,201,045,208,013,071
51624 :173,022,192,201,001,240,229
51630 :006,206,022,192,076,010,174
51636 :200,201,073,208,014,160,012
51642 :063,177,139,073,255,145,014
51648 :139,136,016,247,076,092,130
51654 :200,201,136,208,010,160,089
51660 :063,177,139,153,122,206,040
51666 :136,016,248,201,140,208,135
51672 :013,160,063,185,122,206,197
51678 :145,139,136,016,248,076,214
51684 :092,200,201,145,208,011,061

51690 :162,003,032,061,203,202,129
51696 :200,250,076,092,200,201,243
51702 :017,208,011,162,003,032,167
51708 :091,203,202,208,250,076,002
51714 :092,200,201,029,208,027,247
51720 :160,002,177,139,106,136,216
51726 :136,162,003,177,139,106,225
51732 :145,139,200,202,208,247,137
51738 :200,200,192,066,144,234,038
51744 :076,092,200,201,157,208,198
51750 :029,160,000,177,139,042,073
51756 :200,200,162,003,177,139,157
51762 :042,145,139,136,202,208,154
51768 :247,200,200,200,200,192,015
51774 :064,144,232,076,092,200,102
51780 :201,068,208,014,172,022,241
51786 :192,185,220,055,073,001,032
51792 :153,220,055,076,010,200,026
51798 :173,141,002,201,004,240,079
51804 :062,164,203,185,129,235,046
51810 :205,060,003,240,052,141,031
51816 :060,003,201,049,144,045,094
51822 :201,058,176,041,174,022,014
51828 :192,056,233,049,168,185,231
51834 :050,206,157,250,055,185,001
51840 :041,206,157,024,056,173,017
51846 :141,002,240,014,189,024,232
51852 :056,010,157,024,056,189,120
51858 :250,055,042,157,250,055,187
51864 :076,010,200,201,092,200,171
51870 :003,076,010,200,032,225,192
51876 :202,160,000,076,216,200,250
51882 :169,065,160,205,032,030,063
51888 :171,169,021,141,020,192,122
51894 :169,068,160,205,032,030,078
51900 :171,206,020,192,208,244,205
51906 :162,012,160,033,024,032,105
51912 :240,255,169,000,174,022,036
51918 :192,032,205,189,169,032,001
51924 :032,210,255,096,160,000,197
51930 :177,139,041,127,145,139,218
51936 :096,165,141,133,163,165,063
51942 :142,133,164,169,000,141,211
51948 :038,192,169,003,141,039,050
51954 :192,160,007,169,000,141,143
51960 :042,192,024,177,163,041,119
51966 :015,240,001,056,110,042,206
51972 :192,136,016,242,172,038,032
51978 :192,173,042,192,145,139,125
51984 :165,163,024,105,008,133,102
51990 :163,165,164,105,000,133,240
51996 :164,238,038,192,173,038,103
52002 :192,201,064,208,001,096,028
52008 :206,039,192,208,198,165,024
52014 :163,024,105,016,133,163,138
52020 :165,164,105,000,133,164,015
52026 :076,238,202,160,000,177,143
52032 :139,141,038,192,136,200,142
52038 :200,192,064,208,008,173,147
52044 :038,192,136,136,145,139,094
52050 :096,177,139,136,145,139,146
52056 :076,069,203,160,062,177,067
52062 :139,141,038,192,200,136,172
52068 :136,016,007,173,038,192,150
52074 :200,145,139,096,177,139,234
52080 :200,145,139,076,099,203,206
52086 :169,000,133,251,169,216,032
52092 :133,252,162,004,160,000,067
52098 :152,145,251,136,208,251,249
52104 :230,252,202,208,243,096,087
52110 :120,165,001,041,251,133,085
52116 :001,169,000,133,251,169,103
52122 :032,133,252,169,000,133,105
52128 :253,169,208,133,254,162,059
52134 :008,160,000,177,253,145,141
52140 :251,136,208,249,230,252,218
52146 :230,254,202,208,240,165,197
52152 :001,009,004,133,001,088,164
52158 :160,007,185,100,205,153,232
52164 :008,033,185,108,205,153,120
52170 :008,037,136,016,241,169,041
52176 :024,141,024,208,160,023,020
52182 :169,000,153,000,212,136,116
52188 :016,250,169,015,141,024,067
52194 :212,169,050,141,001,212,243
52200 :169,019,141,005,212,096,106
52206 :169,032,160,000,153,000,240
52212 :008,153,000,009,153,000,055
52218 :010,153,000,011,136,208,000
52224 :241,096,032,038,204,169,012
52230 :067,160,206,032,030,171,160

52236 :032,228,255,201,089,024,073
52242 :240,005,201,078,208,244,226
52248 :056,000,032,038,204,169,019
52254 :083,160,206,032,030,171,200
52260 :040,096,162,015,160,014,011
52266 :024,032,240,255,096,040,225
52272 :088,128,168,208,248,033,153
52278 :144,182,163,163,163,181,026
52284 :017,157,157,157,157,157,094
52290 :182,032,032,032,181,017,030
52296 :157,157,157,157,157,182,015
52302 :175,175,175,181,145,145,050
52308 :000,144,032,032,049,032,117
52314 :032,032,032,050,032,032,044
52320 :032,032,051,032,032,032,051
52326 :032,052,032,032,032,032,058
52332 :053,032,032,032,032,054,087
52338 :032,032,032,032,055,000,041
52344 :032,032,056,032,032,032,000
52350 :032,057,032,032,032,032,087
52356 :049,048,032,032,032,049,118
52362 :049,032,032,032,049,050,126
52368 :032,032,032,049,051,032,116
52374 :032,032,049,052,000,032,091
52380 :049,053,032,032,032,032,130
52386 :049,054,032,032,032,049,154
52392 :055,032,032,032,049,056,168
52398 :032,032,032,049,057,032,152
52404 :032,032,050,048,032,032,150
52410 :032,050,049,000,144,010,223
52416 :083,084,065,082,084,073,151
52422 :078,071,032,080,073,067,087

52428 :084,085,082,069,146,144,046
52434 :032,032,032,032,069,070,229
52440 :060,073,070,071,032,080,106
52446 :073,067,084,085,082,069,170
52452 :032,032,013,029,000,073,231
52458 :067,084,085,082,069,032,141
52464 :078,085,077,066,069,002,185
52470 :032,032,032,032,032,032,182
52476 :079,086,069,082,065,076,197
52482 :076,032,083,080,069,069,155
52488 :068,032,032,032,013,029,214
52494 :083,087,073,004,067,072,224
52500 :073,078,071,032,083,080,181
52506 :069,069,068,032,032,032,072
52512 :032,032,078,079,084,069,150
52518 :032,068,085,082,065,084,198
52524 :073,079,078,032,032,032,114
52530 :000,249,006,249,006,033,081
52536 :007,073,007,013,007,053,216
52542 :007,093,007,147,017,000,077
52548 :013,146,144,029,029,029,202
52554 :029,033,033,033,033,033,012
52560 :033,033,033,033,033,033,022
52566 :033,033,033,033,033,033,028
52572 :033,033,033,033,033,033,034
52578 :033,000,000,126,126,126,253
52584 :126,126,126,000,255,129,098
52590 :129,129,129,129,255,242
52596 :017,028,018,069,146,068,206
52602 :073,084,032,032,032,032,151
52608 :032,032,018,076,146,079,255
52614 :065,068,032,032,032,032,139
52620 :032,032,018,083,146,065,004
52626 :086,069,032,032,032,032,173
52632 :032,032,018,067,146,076,011
52638 :069,065,082,032,065,076,035
52644 :076,032,018,036,146,067,027
52650 :065,084,065,076,079,071,098
52656 :032,032,018,073,146,078,043
52662 :083,069,082,084,032,032,052
52668 :032,032,018,068,146,069,041
52674 :076,069,084,069,032,032,044
52680 :032,032,018,081,146,085,082
52686 :073,084,144,000,017,017,029
52692 :029,069,078,084,069,082,111
52698 :032,070,073,076,069,078,104
52704 :065,077,069,013,000,017,209
52710 :017,032,018,084,146,065,080
52716 :080,069,032,079,082,032,098
52722 :018,068,146,073,083,075,193
52728 :013,013,000,147,005,017,187
52734 :017,000,013,017,032,072,149
52740 :073,084,032,082,069,084,172
52746 :085,082,078,032,084,079,194
52752 :032,067,079,078,084,073,173
52758 :078,085,069,000,068,082,148
52764 :085,077,000,078,079,084,175
52770 :069,000,032,032,032,032,231
52776 :000,097,104,143,048,143,063
```

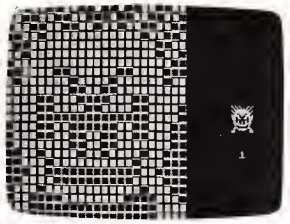
```
52782 :024,210,195,000,008,009,236
52788 :010,011,012,014,015,016,130
52794 :000,036,048,058,065,078,087
52800 :073,046,042,028,065,082,144
52806 :069,032,089,079,085,032,200
52812 :083,085,082,069,063,144,090
52818 :000,032,032,032,032,032,242
52824 :032,032,032,032,032,032,024
52830 :032,032,000,073,048,013,036
52836 :013,032,068,069,086,073,185
52842 :067,069,032,078,079,084,003
52848 :032,080,082,069,083,069,015
52854 :078,084,013,000,000,032,069
52860 :000,004,016,000,008,008,160
52866 :000,016,004,000,032,003,185
52872 :255,192,013,000,176,048,052
52878 :129,012,064,066,002,112,015
52884 :000,014,124,000,062,127,219
52890 :255,254,127,255,254,127,146
52896 :255,254,127,255,254,127,152
52902 :255,254,063,255,252,015,236
52908 :255,240,003,255,192,000,093
52914 :000,000,000,000,000,000,178
52920 :000,000,000,000,004,000,188
52926 :000,006,000,000,005,000,201
52932 :000,004,128,000,004,064,140
52938 :000,004,000,000,004,000,210
52944 :000,004,000,000,004,000,216
52950 :000,004,000,000,004,000,222
52956 :000,004,000,001,228,000,197
52962 :003,244,000,007,252,000,220
52968 :007,248,000,003,240,000,218
52974 :001,224,000,000,000,000,207
52980 :000,000,000,000,000,000,244
52986 :002,013,013,013,013,013,061
```

Program 3: Atari Animator, Part 1

Version by Kevin Mykytyn, Editorial Programmer

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering these listings.

```
IB 0 GRAPHICS 0:POKE 752,1:P
     RINT "{CLEAR}{5 DOWN}
     {12 SPACES}PLEASE WAIT":
     PRINT "{DOWN}
     {7 SPACES}THE SCREEN WI
     LL SLANK "
OL 1 CH=0:FOR A=29000 TO 293
     66:READ S:POKE A,S:CH=C
     H+S:NEXT A:IF CH<>43015
      THEN PRINT "ERROR IN D
     ATA":END
FC 2 RUN "D:PART2"
EO 4 DATA 169,3,141,242,6,16
     9,8,141,243,6
CB 5 DATA 165,88,133,205,165
     ,89,133,206,169,0
DP 6 DATA 141,241,6,169,0,14
     1,240,6,160,7
MJ 7 DATA 56,177,205,201,12,
     240,1,24,110,240
HN 8 DATA 6,136,16,242,173,2
     40,6,174,241,6
IF 9 DATA 157,8,96,32,190,11
     3,173,241,6,24
MF 10 DATA 105,8,141,241,6,2
     01,90,176,47,206
HI 11 DATA 242,6,208,39,169,
     3,141,242,6,32
DH 12 DATA 190,113,32,190,11
     3,173,241,6,56,233
HI 13 DATA 23,141,241,6,206,
     243,6,208,14,169
HO 14 DATA 8,141,243,6,173,2
     41,6,24,105,16
DP 15 DATA 141,241,6,24,144,
     163,104,96,165,205
AA 16 DATA 24,105,8,133,205,
     165,206,105,0,133
HK 17 DATA 206,96,165,20,197
     ,20,240,252,104,169
IB 18 DATA 170,141,49,114,16
     9,34,141,59,114,169
```



"Atari Animator" lets you create cartoons in 128 different colors.

```
KA 19 DATA 154,141,71,114,10
     4,104,10,170,189,139
PI 20 DATA 114,133,203,232,1
     89,139,114,133,204,104
KA 21 DATA 141,242,6,48,20,1
     04,24,109,244,6
ML 22 DATA 141,244,6,173,246
     ,6,109,242,6,141
HO 23 DATA 246,6,24,144,31,7
     3,255,24,105,1
NH 24 DATA 141,242,6,104,141
     ,245,6,173,244,6
LI 25 DATA 56,237,245,6,141,
     244,6,173,246,6
HK 26 DATA 237,242,6,141,246
     ,6,162,3,160,7
HM 27 DATA 177,203,153,170,9
     4,136,16,248,160,15
FJ 28 DATA 177,203,153,42,95
     ,136,192,8,176,246
HN 29 DATA 160,23,177,203,15
     3,170,95,136,192,16
JO 30 DATA 176,246,165,203,2
     4,105,24,133,203,165
CM 31 DATA 204,105,0,133,204
     ,173,49,114,24,105
MP 32 DATA 8,141,49,114,173,
     59,114,105,8,141
ML 33 DATA 59,114,173,71,114
     ,24,105,8,141,71
DI 34 DATA 114,202,208,180,1
     73,246,6,141,1,208
GD 35 DATA 24,105,8,141,2,20
     8,24,105,8,141
PB 36 DATA 3,208,96,8,72,8,7
     2,80,72,152
IF 37 DATA 72,224,72,40,73,1
     12,73,184,73,8
KK 38 DATA 80,80,80,152,80,2
     24,80,40,81,112
KB 39 DATA 81,184,81,8,88,80
     ,88,152,88,224
NO 40 DATA 88,40,89,112,89,1
     84,89
```

Program 4: Atari Animator, Part 2

```
FJ 10 POKE 559,0:GOSUB 1000:
      GOSUB 730:GOSUB 940:GR
      APHICS 0:POKE 752,1:GO
      SUB 280:A=USR(1535)
KJ 20 GOSUB 680:GOSUB 460:GO
      TO 20
JK 30 R=0:TRAP 120:GOSUB 165
      0:OPEN #1,6,0,"D:*.*"
CM 40 INPUT #1,BOX$:PRINT BO
      X$:GOTO 40
PG 50 TRAP 120:X=NUMBER(1):G
      OSUB 160:FA=8A:X=NUMBE
      R(4)+1:GOSUB 160
DP 60 FAH=INT(FA/256):FAL=FA
      -FAH*256:SAH=INT(8A/25
      6):SAL=8A-SAH*256
EC 70 GOSUB 170:IF FN$="" TH
      EN 110
DF 75 IF OEV$="C:" THEN PRIN
      T "{DOWN}PRESS RECORD
      AND PLAY ON TAPE"
LM 80 OPEN #1,8,0,FN$:PUT #1
      ,FAL:PUT #1,FAH:PUT #1
      ,SAL:PUT #1,SAH
IM 90 FOR A=FA TO SA
HP 100 PUT #1,PEEK(A):NEXT A
      :CLOSE #1
PK 110 GOSUB 1670:RETURN
ML 120 TRAP 32767:IF PEEK(19
      5)<>136 THEN PRINT "
      {3 DOWN}SYSTEM ERROR
      ":PEEK(195)
BC 130 PRINT "{DOWN}HIT ANY
      KEY TO CONTINUE":POKE
       764,255
NI 140 IF PEEK(764)=255 THEN
       140
AC 150 GOSUB 1670:CLOSE #1:R
      ETURN
HC 160 Y=X-1:8A=(X-(INT(Y/7)
      *7))*72+(INT(Y/7))*20
      48+1836:RETURN
A8 170 ON$="":FN$="":GOSUB 1
      650:PRINT "{4 DOWN}EN
      TER FILENAME"
KE 180 INPUT ON$:IF ON$="" T
      HEN 240
IK 190 PRINT "{3 DOWN}TAPE O
      R DISK":POKE 764,255
B8 200 IF PEEK(764)=45 THEN
      OEV$="C:":GOTO 230
BM 210 IF PEEK(764)=58 THEN
      OEV$="O:":GOTO 230
FP 220 GOTO 200
E8 230 FN$(3,LEN(ON$)+3)=ON$
      :FN$(1,2)=OEV$
H8 240 RETURN
HB 250 TRAP 120:SOSUS 170:IF
       FN$="" THEN 110
ND 255 IF OEV$="C:" THEN PRI
      NT "{DOWN}PRESS PLAY
      ON TAPE"
MP 260 OPEN #1,4,0,FN$:GET #
      1,FAL:GET #1,FAH:GET
      #1,SAL:GET #1,SAH:FA=
      FAL+256*FAH:8A=SAL+25
      6*SAH
KF 270 FOR A=FA TO 8A+64:GET
       #1,8:POKE A,8:NEXT A
      :SOSUS 1670:CLOSE #1:
      RETURN
OA 280 SOX$=" {3 N}{DOWN}
      {4 LEFT}{8}
      {3 SPACE8}{V}{DOWN}
      {5 LEFT}{8}
      {3 SPACE8}{V}{DOWN}
      {5 LEFT}{8}
      {3 SPACE8}{V}{DOWN}
      {4 LEFT}{3 M}{4 UP}
      {RIGHT}"
OJ 290 PRINT "{CLEAR}"
OO 300 FOR A=0 TO 12 STEP 6:
      POSITION 2,A
MP 310 FOR 8=1 TO 7:PRINT SO
      X$;:NEXT 8
BF 320 NEXT A
BM 330 POSITION 4,5:PRINT "1
      {4 SPACES}2
      {4 SPACES}3
      {4 SPACES}4
      {4 SPACES}5
      {4 SPACES}6
      {4 SPACES}7"
DJ 340 POSITION 4,11:PRINT "
      8{4 SPACES}9
      {4 SPACES}10
      {3 SPACES}11
      {3 SPACES}12
      {3 SPACES}13
      {3 SPACES}14"
KM 350 POSITION 4,17:PRINT "
      15{3 SPACES}16
```

```
      (3 BPACEB)17
      (3 BPACES)18
      (3 BPACEB)19
      (3 BPACES)20
      (3 BPACES)21"
CH 360 E=-9:FDR B=0 TO 30 ST
      EP 5:E=E+9
KE 370 FDR A=1 TO 13 BTEP 6:
      CHAR=33
DD 380 FOR O=0 TD 2:PDSITIDN
      3+B,A+D
FI 390 FOR C=0 TD 2:PRINT CH
      R$(CHAR+C+E);
OE 400 NEXT C:PRINT "(DOWN)
      (3 LEFT)";:CHAR=CHAR+
      3
IN 410 NEXT D:NEXT A:NEXT B
HK 420 PDBITION 1,19:PRINT "
      START PICTURE
      (7 BPACES)END PICTURE
      "
ED 430 PDSITION 1,20:PRINT "
      PICTURE NUMBER
      (6 SPACEB)OVERALL SPE
      ED"
PJ 440 PDSITIDN 1,21:PRINT "
      BWITCHING SPEEO "
HJ 450 RETURN
NI 460 JDY=BTICK(0)
BK 470 ON JOY-4 GDTD 560,560
      ,480,560,560,560,500,
      560,520,540,560
EJ 480 IF NUMBER(ARRPOB)=21
      THEN NUMBER(ARRPOS)=0
CB 490 NUMBER(ARRPOB)=NUMBER
      (ARRPDB)+1:GOTO 560
EE 500 IF NUMBER(ARRPDB)=1 T
      HEN NUMBER(ARRPDS)=22
CB 510 NUMBER(ARRPOB)=NUMBER
      (ARRPOS)-1:GDTO 560
DI 520 GDBUB 660:IF ARRPDS=5
       THEN ARRPOS=0
ON 530 ARRPOS=ARRPOB+1:GOTO
      560
DH 540 GDSUS 660:IF ARRPDS=1
       THEN ARRPOS=6
OD 550 ARRPOS=ARRPDS-1
DF 560 POBITIDN CDDRD(ARRPDS
      ,1),CDDRO(ARRPOB,2)
PRINT ">"
KL 570
BM 580 IF BTRIG(0)=0 THEN GO
      BUB 1540
KN 590 KEY=PEEK(764):PDKE 76
      4,255
CC 600 IF KEY=42 THEN GDBUB
      1100
HI 610 IF KEY=62 THEN GDSUB
      50
HO 620 IF KEY=0 THEN GDSUB 2
      50
WH 630 IF KEY=5B THEN GOSUB
      30
PD 640 IF KEY=1S THEN FOR A=
      0 TO 511:POKE A+CHAR1
      ,0:POKE A+CHAR2,0:PDK
      E A+CHAR3,0:NEXT A
WL 650 RETURN
DG 660 PDBITION CODRD(ARRPDS
      ,1),CDDRO(ARRPDB,2)
II 670 PRINT " ":RETURN
AL 680 FOR A=1 TO 4
HH 690 POBITIDN CODRD(A,1)+1
      6+(A<4),CODRO(A,2)
PD 700 PRINT NUMBER(A);" "
ON 710 NEXT A:POSITION 36,20
      :PRINT NUMBER(A)-10;"
      "
HJ 720 RETURN
OB 730 REBTDRE 760:A=1535
LO 740 READ S:IF S=256 THEN
      A=USR(1535):RETURN
NN 750 POKE A,S:A=A+1:GDTD 7
      40
OI 760 DATA 104,173,48,2,141
      ,203,0
CK 770 DATA 173,49,2,141,204
      ,0
JH 780 DATA 160,2B,169,2,145
      ,203
NO 790 DATA 136,192,5,20B,24
      9,162
BN 800 DATA 5,1B9,102,6,16B,
      177
LL 810 OATA 203,9,12B,145,20
      3,202
GA 820 OATA 16,243,169,64,14
      1,0
JC B30 DATA 2,169,6,141,1,2
MO B40 DATA 169,255,141,10B,
      6,173
PI B50 DATA 11,212,20B,251,1
      69,192
JI 860 DATA 141,14,212,96,72
      ,13B
JO 870 OATA 72,23B,10B,6,173
      ,108
AE BB0 DATA 6,201,6,20B,5,16
      9
BG B90 DATA 0,141,10B,6,170,
      1B9
FG 900 DATA 96,6,141,10,212,
      141
IK 910 DATA 9,212,104,170,10
      4,64
JK 920 DATA 72,224,B0,224,BB
      ,224
FH 930 OATA 3,B,11,14,17,20,
      256
FL 940 ORIGINAL=57344:CHAR1=
      1B432:CHAR2=20480:CHA
      R3=22528:CHAR0=24576
EC 950 FDR A=ORIGINAL TO DRI
      GINAL+1024
GH 960 D=A-DRIGINAL:V=PEEK(A
      )
MG 970 POKE CHAR1+O,V:PDKE C
      HAR2+D,V:PDKE CHAR3+O
      ,V:PDKE CHAR0+D,V:NEX
      T A
OD 9B0 REBTDRE 990:FDR A=246
      64 TD 246B7:READ B:PO
      KE A,S:NEXT A:RETURN
EG 990 DATA 48,4B,4B,255,255
      ,4B,48,4B,0,126,126,1
      26,126,126,126,0,255,
      129,129,129,129,129,1
      29,255
II 1000 DIM BOX$(50):DIM NUM
      BER(5):DIM B$(5):DIM
       SINARY$(64):DIM FN$
      (15):DIM DEV$(2):DIM
       DN$(17):DIM CODRD(5
      ,2)
FP 1010 POKE 752,1:RESTDRE 1
      030:FDR A=1 TO 5:NUM
      BER(A)=1:FOR S=1 TD
      2
HI 1020 READ C:CDDRD(A,S)=C:
      NEXT S:NEXT A
LJ 1030 OATA 0,19,0,20,0,21,
      20,19,20,20
KE 1040 ARRPOB=1:RESTORE 106
      0:FOR A=1 TO 64:READ
       B
CH 1050 BINARY$(A,A)=CHR$(B+
      1+44):NEXT A
BO 1060 DATA 1,1,1,1,1,1,1,0
      ,1,1,0,1,1,1,0,0,1,0
      ,1,1,1,0,1,0,1,0,0,1
      ,1,0,0,0,0,1,1,1,0,1
      ,1,0,0,1,0,1,0,1,0,0
HK 1070 OATA 0,0,1,1,0,0,1,0
      ,0,0,0,1,0,0,0,0
PJ 1080 DL1=PEEK(560)+256*PE
      EK(561):ODL=PEEK(DLI
      +4):DDH=PEEK(DLI+5)
KK 1090 RETURN
DL 1100 GDSUB 1110:GDTD 1200
PD 1110 NU=NUMBER(2)-1:PDKE
      756,96
NH 1120 SA=(NUMBER(2)-(INT(N
      U/7)*7))*72+(INT(NU/
      7))*204B+1836B
GB 1130 GDSUB 1650
EJ 1140 R=0:S=0:NR=S:FOR A=S
      A TD BA+71:P=PEEK(A)
LE 1150 HB=INT(P/16)+1:LB=P-
      16*(HB-1)+1
OB 1160 PDSITION S*S,R:PRINT
       BINARY$(HB*4-3,HB*4
      );BINARY$(LS*4-3,LB*
      4);
EN 1170 R=R+1:IF R=NR THEN R
      =NR-S:B=B+1:IF S=3 T
      HEN R=R+B:NR=NR+B:B=
      0
EK 1180 NEXT A
KL 1190 RETURN
EC 1200 PDBITIDN 31,16:PRINT
       NUMBER(2)
ND 1210 FDR A=0 TD 2:FDR B=0
      TD 2
LJ 1220 PDBITION 30+B,11+A:P
      RINT CHR$(33+B+A*3)
AD 1230 NEXT B:NEXT A
CF 1240 A=UBR(29000):IF FL=0
      THEN LDCATE DX,DY,Z
IJ 1250 DX=DX:DY=DY:PDBITIDN
      DX,DY:PRINT "+";
FI 1260 JOY=BTICK(0)-4
OG 1270 DN JOY GDTO 1280,129
      0,1300,1360,1310,132
      0,1330,1360,1340,135
      0,1360
OG 1280 DX=DX+1:DY=DY+1:GDTO
      1360
OJ 1290 DX=DX+1:OY=OY-1:GDTO
      1360
NC 1300 DX=DX+1:GOTD 1360
OC 1310 DX=DX-1:DY=OY+1:BDTD
      1360
OF 1320 DX=DX-1:OY=DY-1:GDTD
      1360
NH 1330 DX=DX-1:GDTD 1360
NI 1340 DY=OY+1:GOTD 1360
NL 1350 DY=OY-1:GDTO 1360
NA 1360 IF OX>23 THEN DX=0
ND 1370 IF DY>23 THEN OY=0
NA 1380 IF OX<0 THEN DX=23
ND 1390 IF OY<0 THEN OY=23
NL 1400 IF STRIG(0)=0 ANO FL
      =0 THEN FL=1:Z=(Z+44
      )$1+44
OH 1410 IF BTRIG(0)<>0 THEN
      FL=0
GF 1420 POBITIDN DX,OY:PRINT
       CHR$(Z);
NH 1430 KEY=PEEK(764):PDKE 7
      64,255
DL 1440 IF KEY=1B THEN FDR A
      =0 TD 23:FDR B=0 TO
      23:POBITIDN A,B:PRIN
      T CHR$(45);:NEXT S:N
      EXT A
NL 1450 IF KEY=42 THEN 1510
IL 1460 IF KEY=6 AND NUMBER(
      2)<21 THEN GOSUB 152
      0:NUMBER(2)=NUMBER(2
      )+1:GOTO 1100
IN 1470 IF KEY=14 AND NUMBER
      (2)>1 THEN GDSUB 152
      0:NUMBER(2)=NUMBER(2
      )-1:GOTO 1100
IN 1480 IF KEY=61 THEN SHP=N
      UMBER(2)
BC 1490 IF KEY=10 THEN TEMP=
      NUMBER(2):TEMP2=SA:N
      UMBER(2)=SHP:GDSUS 1
      110:NUMBER(2)=TEMP:S
```

```
      A=TEMP2:GOTO 1200
M6 1500 GOTO 1240
L8 1510 GOSUB 1670:GOSUB 152
        0:RETURN
KA 1520 FOR A=0 TO 71:POKE 8
        A+A,PEEK(24584+A):NE
        XT A
KJ 1530 RETURN
NJ 1540 IF STRIG(0)=0 THEN 1
        540
DD 1550 GOSUB 1650:POKE 559,
        46:POKE 623,1:POKE 7
        05,8:POKE 706,8:POKE
        707,8:POKE 53277,3:
        POKE 54279,92
KF 1560 SP=NUMBER(1):EP=NUMB
        ER(4):DIR=SGN(EP-SP)
        :S8=NUMBER(3):D8=NUM
        BER(5)-1Ø
DB 1570 S=A88(D8)/3%256:IF D
        8<0 THEN Q=255-INT(S
        /256):R=S-(255-Q)%25
        6:S=Q%256+R
AA 1580 D8=8
IH 1590 FOR PN=SP TO EP STEP
        DIR
CJ 1600 A=USR(29132,PN,D8)
PE 1610 TD=TD-1:IF TD<0 THEN
        TD=22-8Ø:NEXT PN:GO
        TO 1590
FH 1620 KEY=PEEK(764):POKE 7
        05,KEY:POKE 706,KEY:
        POKE 707,KEY
AD 1630 IF STRIG(0)<>0 THEN
        1600
JP 1640 FOR A=53248 TO 53251
        :POKE A,220:NEXT A:G
        OSUB 1670:RETURN
MC 1650 POKE 54286,64:POKE 8
        8,0:POKE 89,100:POKE
        DLI+4,0:POKE DLI+5,
        100
HH 1660 T=PEEK(106):POKE 106
        ,104:PRINT CHR$(125)
        :POKE 106,T:RETURN
JJ 1670 A=USR(1535):POKE DLI
        +4,DDL:POKE DLI+5,DD
        H:POKE 88,DDL:POKE 8
        9,DDH:RETURN
```

Program 5: Apple ANIMATOR2 Maker

Version by Tim Victor, Editorial Programmer

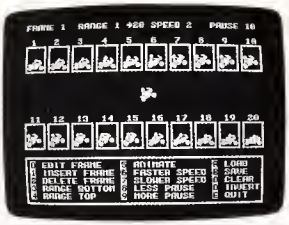
Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering these listings.

```
26 80 FOR I = 126 * 256 TO 1 + 1
   455: READ A: POKE I,A: NEX
   T
F5 90 PRINT CHR$ (4);"BSAVE ANIM
   ATOR2,A$7E00,L$580": END
17 100 DATA 0,0,0,0,0,0,0,0,0,0,
   0
82 110 DATA 0,0,0,0,0,0,216,120,13
   3,69,134,70
78 120 DATA 132,71,166,7,10,10,1
   76,4,16,62,48
13 130 DATA 4,16,1,232,232,10,13
   4,27,24,101,6
55 140 DATA 133,26,144,2,230,27,
   165,40,133,8,165
63 150 DATA 41,41,3,5,230,133,9,
   162,8,160,0
78 160 DATA 177,26,36,50,48,2,73
   ,127,164,36,145
9C 170 DATA 8,230,26,208,2,230,2
   7,165,9,24,105
86 180 DATA 4,133,9,202,208,226,
   165,69,166,70,164
8B 190 DATA 71,88,76,240,253,128
   ,128,128,128,190,128
55 200 DATA 128,128,128,128,128,
   190,128,190,128,128,128
11 210 DATA 188,230,176,152,128,
   152,128,128,188,230,246
63 220 DATA 238,230,188,128,128,
   152,156,152,152,152,188
E8 230 DATA 128,128,188,230,176,
   140,230,254,128,128,188
7E 240 DATA 230,176,224,230,188,
   128,128,176,184,180,254
EC 250 DATA 176,176,128,128,254,
   134,190,224,230,188,128
8A 260 DATA 128,188,134,190,230,
   230,188,128,128,254,224
81 270 DATA 176,152,140,140,128,
   128,188,230,188,230,230
F3 280 DATA 188,128,128,188,230,
   230,252,176,152,128,128
98 290 DATA 152,176,254,254,176,
   152,128,128,190,190,190
21 300 DATA 190,190,190,128,0,0,
   0,0,0,0,0
18 310 DATA 0,0,0,0,0,0,0,0,0,0,
   0
1D 320 DATA 0,0,0,0,0,0,0,0,0,0,
   0
A7 330 DATA 0,0,0,128,128,152,18
   8,188,152,128,128
64 340 DATA 128,252,230,230,254,
   230,230,128,128,190,230
A2 350 DATA 230,190,230,254,128,
   128,188,230,134,134,230
F1 360 DATA 190,128,128,190,230,
   230,230,230,190,128,128
AD 370 DATA 254,134,134,190,134,
   254,128,128,254,134,134
F4 380 DATA 190,134,134,128,128,
   188,230,134,246,230,190
8A 390 DATA 128,128,230,230,230,
   254,230,230,128,128,152
3C 400 DATA 152,152,152,152,152,
   128,128,224,224,224,224
87 410 DATA 230,188,128,128,230,
   230,182,158,230,230,128
2B 420 DATA 128,134,134,134,134,
   134,254,128,128,230,254
97 430 DATA 230,230,230,230,128,
   128,190,230,230,230,230
69 440 DATA 230,128,128,188,230,
   230,230,230,188,128,128
88 450 DATA 190,230,230,190,134,
   134,128,128,188,230,230
8A 460 DATA 230,182,236,128,128,
   190,230,230,190,230,230
82 470 DATA 128,128,188,230,140,
   176,230,190,128,128,254
5D 480 DATA 152,152,152,152,152,
   128,128,230,230,230,230
5D 490 DATA 230,190,128,128,230,
   230,230,230,230,152,128
3C 500 DATA 128,230,230,230,230,
   254,230,128,128,230,230
27 510 DATA 230,188,230,230,128,
   128,230,230,230,188,152
F4 520 DATA 152,128,128,254,176,
   152,140,134,254,128,0
94 530 DATA 0,0,0,0,0,0,0,160,0,
   162,7
D7 540 DATA 169,196,133,254,169,
   132,133,255,177,254,73
44 550 DATA 127,145,254,200,208,
   247,230,255,202,208,242
38 560 DATA 96,0,0,0,0,0,76,21,1
   28,76,85
85 570 DATA 128,76,130,128,76,23
   9,128,76,6,130,76
14 580 DATA 148,129,76,210,129,1
   69,2,141,180,131,169
82 590 DATA 24,141,181,131,32,31
   ,130,176,48,32,71
3B 600 DATA 131,176,43,32,156,13
   1,176,38,173,178,131
4C 610 DATA 133,252,173,179,131,
   133,253,32,223,130,32
4D 620 DATA 95,130,32,168,130,23
   8,184,131,165,252,24
8B 630 DATA 105,3,133,252,144,2,
   230,253,206,181,131
D4 640 DATA 208,228,96,169,2,141
   ,180,131,169,24,141
E8 650 DATA 181,131,32,71,131,17
   6,29,32,156,131,176
DE 660 DATA 24,160,3,169,0,153,1
   96,131,136,16,250
CD 670 DATA 32,223,130,32,168,13
   0,238,184,131,206,181
52 680 DATA 131,208,242,96,165,5
   6,164,57,201,186,208
FE 690 DATA 4,192,128,240,14,141
   ,190,131,140,191,131
EC 700 DATA 169,186,133,56,160,1
   28,132,57,32,121,129
83 710 DATA 169,0,141,194,131,32
   ,31,130,176,13,173
42 720 DATA 178,131,141,197,128,
   173,179,131,141,198,128
DC 730 DATA 96,32,6,130,0,145,40
   ,169,48,141,0
3F 740 DATA 2,141,1,2,173,255,25
   5,201,100,144,7
2A 750 DATA 233,100,238,0,2,208,
   245,201,10,144,7
8E 760 DATA 233,10,238,1,2,208,2
   45,105,48,141,2
FC 770 DATA 2,162,3,169,141,238,
   197,128,208,3,238
F7 780 DATA 198,128,96,32,121,12
   9,169,255,141,194,131
37 790 DATA 32,31,130,176,23,173
   ,178,131,141,84,129
CC 800 DATA 173,179,131,141,85,1
   29,169,196,141,110,129
98 810 DATA 169,131,141,111,129,
   96,32,6,130,0,44
54 820 DATA 194,131,48,3,76,240,
   253,201,176,144,4
32 830 DATA 201,186,144,70,72,14
   0,195,131,56,173,110
3D 840 DATA 129,233,196,240,53,1
   41,182,131,169,0,168
22 850 DATA 201,26,176,214,10,14
   1,68,129,10,10,105
82 860 DATA 255,176,204,121,196,
   131,56,233,176,200,204
BB 870 DATA 182,131,208,230,141,
   255,255,238,84,129,208
31 880 DATA 3,238,85,129,169,196
   ,141,110,129,169,131
ED 890 DATA 141,111,129,172,195,
   131,104,96,141,255,255
9D 900 DATA 238,110,129,208,3,23
   8,111,129,96,165,54
4F 910 DATA 164,55,201,23,208,4,
   192,129,240,14,141
CC 920 DATA 192,131,140,193,131,
   169,23,133,54,160,129
E4 930 DATA 132,55,96,32,31,130,
   176,56,173,178,131
63 940 DATA 141,198,129,208,3,20
   6,179,131,206,198,129
1E 950 DATA 173,179,131,141,204,
   129,169,227,133,252,169
82 960 DATA 142,133,253,160,0,17
   7,252,160,72,145,252
27 970 DATA 165,252,208,2,198,25
   3,198,252,169,255,197
93 980 DATA 252,208,234,169,255,
   197,253,208,228,96,32
89 990 DATA 31,130,176,46,173,17
   8,131,133,252,173,179
6C 1000 DATA 131,133,253,169,228
   ,141,250,129,169,142,141
29 1010 DATA 0,130,160,72,177,25
   2,160,0,145,252,230
E7 1020 DATA 252,208,2,230,253,1
   69,255,197,252,208,236
6D 1030 DATA 169,255,197,253,208
   ,230,96,173,190,131,172
```

```
FC 1040 DATA 191,131,240,4,133,5
   6,132,57,173,192,131
49 1050 DATA 172,193,131,240,4,1
   33,54,132,55,96,169
C8 1060 DATA 196,141,178,131,169
   ,132,141,179,131,32,165
9B 1070 DATA 131,201,21,144,1,96
   ,141,176,131,169,0
B6 1080 DATA 141,177,131,160,3,3
   2,64,130,160,3,173
AB 1090 DATA 176,131,10,46,177,1
   31,136,208,249,141,176
E6 1100 DATA 131,24,109,178,131,
   141,178,131,173,179,131
44 1110 DATA 109,177,131,141,179
   ,131,24,96,172,180,131
C4 1120 DATA 200,140,182,131,169
   ,0,153,196,131,136,177
2E 1130 DATA 252,153,196,131,136
   ,16,248,173,196,131,9
78 1140 DATA 127,141,183,131,172
   ,187,131,240,21,162,0
72 1150 DATA 14,196,131,189,196,
   131,10,62,197,131,232
BB 1160 DATA 236,182,131,208,243
   ,136,208,235,172,182,131
C9 1170 DATA 185,196,131,9,128,4
   5,183,131,153,196,131
5C 1180 DATA 136,16,242,96,172,1
   87,131,185,209,130,172
77 1190 DATA 182,131,49,254,25,1
   96,131,145,254,136,185
BE 1200 DATA 196,131,145,254,136
   ,208,248,172,187,131,185
4D 1210 DATA 216,130,160,0,49,25
   4,13,196,131,145,254
0D 1220 DATA 96,127,126,124,120,
   112,96,64,0,1,3
6C 1230 DATA 7,15,31,63,173,184,
   131,41,63,168,185
FA 1240 DATA 7,131,5,230,133,255
   ,173,184,131,41,8
9A 1250 DATA 240,2,169,128,24,44
   ,184,131,112,4,16
BF 1260 DATA 4,105,40,105,40,109
   ,186,131,133,254,96
78 1270 DATA 0,4,8,12,16,20,24,2
   8,0,4,8
69 1280 DATA 12,16,20,24,28,1,5,
   9,13,17,21
26 1290 DATA 25,29,1,5,9,13,17,2
   1,25,29,2
B0 1300 DATA 6,10,14,18,22,26,30
   ,2,6,10,14
7D 1310 DATA 18,22,26,30,3,7,11,
   15,19,23,27
49 1320 DATA 31,3,7,11,15,19,23,
   27,31,169,0
7B 1330 DATA 141,186,131,141,187
   ,131,32,165,131,141,185
DB 1340 DATA 131,192,1,144,18,24
   0,1,96,201,24,144
E9 1350 DATA 1,96,169,36,141,186
   ,131,169,4,141,187
CA 1360 DATA 131,169,0,141,188,1
   31,169,224,141,189,131
F2 1370 DATA 173,185,131,205,189
   ,131,144,4,237,189,131
B7 1380 DATA 56,46,188,131,78,18
   9,131,144,239,24,109
6E 1390 DATA 187,131,141,187,131
   ,24,173,188,131,109,186
AC 1400 DATA 131,141,186,131,24,
   96,32,165,131,141,184
F6 1410 DATA 131,201,192,96,32,1
   77,0,32,5,225,165
B0 1420 DATA 161,164,160,96
```

Program 6: Apple Animator

```
FD 100 DIM E%(2,23),M$(8),CM$(7)
   ,MM$(14):D$ = CHR$ (4): P
   RINT D$;"BLOAD ANIMATOR2"
```



In this demonstration of "Apple Animator," a motorcyclist performs a daring wheelstand.

```
AF 110 FOR I = 0 TO 8: READ M$(1
   ): NEXT : FOR I = 0 TO 7:
    READ CM$(I): NEXT : FOR
   I = 0 TO 14: READ MM$(1):
   NEXT
46 120 GOSUB 530
D6 130 A$ = "": FOR 1 = 1 TO 72:
   A$ = A$ + "0 ":A = FRE (0
   ): NEXT
16 140 GOSUB 650: GOSUB 590
16 150 F = 1:AB = 1:AE = 20:AS =
   0:AP = 10
B5 160 ONERR GOTO 1390
6C 170 GOSUB 710: IF MQ = 1 THEN
    HOME : TEXT : END
44 180 C = PEEK (49152): IF C <
   128 THEN 180
45 190 POKE 49168,0: IF C = 136
   THEN F = F - 1 + 20 * (F
   = 1): GOTO 170
CF 200 IF C = 149 THEN F = F + 1
   - 20 * (F = 20): GOTO 17
   0
69 210 C = C - 175: IF C < 1 OR
   C > 22 THEN 180
CB 220 IF C < 11 THEN 250
75 230 IF C < 18 THEN 180
98 240 C = C - 7
6A 250 MQ = 0: ON C GOSUB 260,11
   20,1210,1500,1510,1520,16
   60,1670,1680,1690,1370,12
   50,1470,1460,1440:T = FRE
   (0): GOTO 170
02 260 GDSUB 560: GOSUB 700: VTA
   B 19: HTAB 14: PRINT "EDI
   TING BOX ";A
51 270 PRINT "PRESS ESC TO CANCE
   L": HTAB 7: PRINT "RETURN
    FOR SAME": PRINT : PRINT
   "STORE RESULT IN BOX ";:
   R$ = ""
98 280 XC = 21 + LEN (R$): VTAB
   23: HTAB XC: PRINT ";";
77 290 C = PEEK (49152): IF C <
   128 THEN 290
25 300 POKE 49168,0: IF C = 141
   OR C = 155 THEN VTAB 23:
   HTAB XC: PRINT " ";: GOTO
    360
3A 310 IF C < > 136 AND C < > 25
   5 THEN 340
AE 320 VTAB 23: HTAB XC: PRINT "
   ";: IF LEN (R$) < 2 THEN
   R$ = "": GOTO 280
46 330 R$ = LEFT$ (R$, LEN (R$)
   - 1): GOTO 280
AD 340 IF C < 176 OR C > 185 THE
   N 290
93 350 VTAB 23: HTAB XC: PRINT C
   HR$ (C - 128);:R$ = R$ +
   CHR$ (C - 128):Q = FRE (0
   ): GOTO 280
4E 360 IF C = 155 THEN GDSUB 660
   : RETURN
F6 370 IF R$ = "" THEN AA = A: G
   OTO 390
86 380 AA = VAL (R$): IF AA > 20
    THEN GOSUB 660: RETURN
9A 390 HGR2 : HOME : GOSUB 1070
F2 400 XP = 184:YP = 44:OX = 65:
   DY = 80: GOSUB 980
3A 410 VTAB 7: FOR Q = 0 TO 8: H
   TAB 28: INVERSE : PRINT M
   ID$ ("123456789",Q + 1,1)
   ;: NORMAL : PRINT " ";M$(
   Q): NEXT
BC 420 CALL 32768,A,206,12:XC =
   0:YC = 0:GF = 0: GOSUB 87
   0
44 430 IF GF THEN GOSUB 590: RET
   URN
32 440 SC = INT (YC / 8):SA = 10
   24 + YC * 128 - SC * 984
   + XC:CC$ = " "
B4 450 OC$ = CC$:CC$ = CHR$ ( PE
   EK (SA) - 128): HTAB XC +
    1: VTAB YC + 1: PRINT OC
   $;
C4 460 C = PEEK (49152): IF C <
   128 THEN T = FRE (0): GOT
   O 450
2E 470 POKE 49168,0: IF DC$ = "
   " THEN HTAB XC + 1: VTAB
   YC + 1: PRINT CC$;
91 480 FOR Q = 1 TO 13: IF C < >
    ASC ( MID$ ("JLIK1234567
   89",Q,1)) + 128 THEN NEXT
B6 490 ON Q GOSUB 750,760,770,78
   0,840,850,870,880,890,990
   ,1020,1050,1060: GOTO 430
B8 500 I = A * 28 - 24 - 280 * (
   A > 10):J = 26 + 80 * (A
   > 10): CALL 32768,A,I,J
EB 510 X = FRE (0): RETURN
AD 520 FOR A = 1 TO 20: GOSUB 50
   0: NEXT : RETURN
17 530 POKE 6,0: POKE 7,126: IF
   PEEK (48640) = 76 THEN 55
   0
6B 540 POKE 54,16: POKE 55,126:
   CALL 1002: RETURN
3E 550 PRINT : PRINT CHR$ (4);"P
   R#A$7E10": RETURN
8B 560 A = F: RETURN
E3 570 GOSUB 580: HTAB 1: INPUT
   "WHICH BOX?";A:A = INT (A
   ): IF A < 1 OR A > 20 THE
   N 570
2F 580 VTAB 22: HTAB 1: PRINT SP
   C( 39): RETURN
6F 590 HOME : HGR2 : HCOLOR= 3
8A 600 FOR J = 25 TO 105 STEP 80
   : FOR I = 3 TO I + 9 * 28
    STEP 28: FOR P = 0 TO 1
9D 610 HPLOT I - P,J - P TO I +
   22 + P,J - P TO I + 22 +
   P,J + 25 + P TO I - P,J +
    25 + P TO I - P,J - P
F8 620 NEXT : NEXT : NEXT
1B 630 FOR J = 0 TO 1: FOR I = 1
   TO 10: HTAB I * 4 - 2: V
   TAB J * 10 + 3: PRINT I +
   J * 10: NEXT : NEXT
9D 640 GOSUB 520: GOTO 660
B7 650 POKE 242,0: CALL 32777,0:
    FOR I = 0 TO 20: PRINT A
   $: NEXT : CALL 32780: RET
   URN
3A 660 GOSUB 700:XP = 2:YP = 140
   :DX = 275:DY = 48: GOSUB
   980
AA 670 VTAB 19: FOR I = 0 TO 4:
   HTAB 2: INVERSE : PRINT I
   ;: NORMAL : PRINT " ";MM$(
   I);: HTAB 17: INVERSE :
   PRINT I + 5;
DB 680 NORMAL : PRINT " ";MM$(I
   + 5);: HTAB 32: INVERSE :
    PRINT CHR$ (65 + I);: NO
```

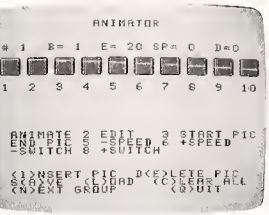
```
      RMAL : PRINT " ";MM$(I +
      10): NEXT
28 690 RETURN
83 700 HTAB 1: VTAB 1B: FOR Q =
      1 TO 7: PRINT SPC( 40): N
      EXT : RETURN
87 710 VTAB 1: HTAB 1: PRINT "FR
      AME ";F;" ";
76 720 HTAB 10: PRINT "RANGE ";A
      B;" ";: HTAB 1B: PRINT ":
      ";AE;" ";
FD 730 HTAB 22: PRINT "SPEED ";A
      S;"  ";: HTAB 32: PRINT "
      PAUSE ";AP;" ";
1F 740 RETURN
1B 750 XC = XC - (XC > 0): GOTO
      790
47 760 XC = XC + (XC < 20): GOTO
      790
6B 770 YC = YC - (YC > 0): GOTO
      790
5B 780 YC = YC + (YC < 23): GOTO
      790
D1 790 XB = INT (XC / 7):PM = 2
      ^ (XC - 7 * XB):T = INT (
      E%(XB,YC) / PM): ON OF GO
      TO 800,B20: RETURN
BB B00 IF T = 2 * INT (T / 2) TH
      EN E%(XB,YC) = E%(XB,YC)
      + PM: HTAB XC + 1: VTAB Y
      C + 1: PRINT ";";
1A B10 RETURN
5B B20 IF T < > 2 * INT (T / 2)
      THEN E%(XB,YC) = E%(XB,YC
      ) - PM: HTAB XC + 1: VTAB
      YC + 1: PRINT ".";
1E B30 RETURN
C8 B40 QF = 1: RETURN
5B B50 OF = 1: GOSUB B60: HTAB 2
      9: VTAB 8: PRINT "@";: GO
      TO 790
B5 B60 VTAB B: FOR I = 1 TO 3: H
      TAB 29: PRINT " ": NEXT :
      RETURN
7B B70 OF = 0: GOSUB B60: HTAB 2
      9: VTAB 9: PRINT "@";: RE
      TURN
CD BB0 OF = 2: GOSUB B60: HTAB 2
      9: VTAB 10: PRINT "@";: G
      OTO 790
D3 B90 XP = 149:YP = 140:OX = 12
      B:DY = 40: GOSUB 9B0
4B 900 FOR I = 0 TO 3: VTAB 19 +
      I: FOR J = 0 TO 1: HTAB
      23 + 9 * J: INVERSE : PRI
      NT I + J * 4;: NORMAL : P
      RINT " ";CM$(I + J * 4);:
      NEXT : NEXT
DD 910 C = PEEK (49152): IF C <
      128 THEN 910
89 920 VTAB 1B: FOR I = 0 TO 5:
      HTAB 22: PRINT SPC( 19):
      NEXT : POKE 4916B,0:C = C
      - 176
E1 930 IF C < 0 OR C > 7 THEN RE
      TURN
22 940 T = C - 4 * INT (C / 4):P
      0 = 42 * T + (T > 1):P1 =
      P0: IF P0 = 42 OR P0 = B
      5 THEN P1 = 127 - P0
8B 950 IF C > 3 THEN P0 = P0 + 1
      2B:P1 = P1 + 12B
2B 960 T = T + (T > 2): FOR I =
      0 TO 23: VTAB I + 1: HTAB
      1:E%(0,I) = P0:E%(1,I) =
      P1:E%(2,I) = P0
52 970 FOR J = 1 TO 10: PRINT MI
      D$ ("..;.;",T + 1,2);: N
      EXT : PRINT MID$ (".;",(T
      > 1) + 1,1);: NEXT : RET
      URN
44 9B0 FOR P = 0 TO 1: HPLOT XP
      + P,YP + P TO XP + OX - P
      ,YP + P TO XP + DX - P,YP
      + OY - P TO XP + P,YP +
      OY - P TO XP + P,YP + P:
      NEXT : RETURN
7C 990 QF = 1: GOSUB 1000: RETUR
      N
7F 1000 POKE 242,0: CALL 32777,A
      A
7D 1010 FOR I = 0 TO 23: FOR J =
      0 TO 2: PRINT E%(J,I):
      NEXT : NEXT : CALL 32780
      : RETURN
F9 1020 HOME : FOR I = 0 TO 23:
      FOR J = 0 TO 2:O = 127 -
      E%(J,I): IF O < 0 THEN
      O = O + 256
9E 1130 GOSUB 700: IF A < 20 THE
      N CALL 32783,A
B6 1140 POKE 242,0: CALL 32777,A
      : PRINT A$: CALL 327B0:
      FOR A = A TO 20: GOSUB 5
      00: NEXT
F0 1150 GOSUB 660: RETURN
ED 1160 PRINT " -REALLY/";
AB 1170 C = PEEK (49152): IF C <
      12B THEN 1170
1F 11B0 POKE 4916B,0: IF C = 206
      THEN PRINT "NO": RETURN
B2 1190 IF C = 217 THEN PRINT "Y
      ES": RETURN
6C 1200 GOTO 1170
78 1210 GOSUB 560: GOSUB 700: VT
      AB 19: HTAB 10: PRINT "D
      ELETE BOX ";A$: GOSUB 11
      60: IF C = 206 THEN 1240
1E 1220 GOSUB 700: IF A < 20 THE
      N CALL 327B6,A
AF 1230 POKE 242,0: CALL 32777,2
      0: PRINT A$: CALL 32780:
      FOR A = A TO 20: GOSUB
      500: NEXT
EE 1240 GOSUB 660: RETURN
98 1250 F$ = "SAVE": GOSUB 1270:
      IF LEN (N$) < > 0 THEN
      PRINT : PRINT O$;"BSAVE
      ";N$;",A$B4C4,L$5EB"
F6 1260 GOTO 590
52 1270 HOME : TEXT : VTAB 2: PR
      INT "ESC TO CANCEL, RETU
      RN FOR CATALOG"
A1 12B0 PRINT : PRINT F$;" FILEN
      AME:";:N$ = ""
49 1290 GOSUB 1320: IF C$ = CHR$
      (27) THEN N$ = "": RETU
      RN
CE 1300 IF N$ = "" THEN PRINT :
      PRINT O$;"CATALOG": GOTO
      12B0
DB 1310 RETURN
AC 1320 T = FRE (0): GET C$: IF
      C$ = CHR$ (13) OR C$ = C
      HR$ (27) THEN RETURN
BA 1330 IF C$ < > CHR$ (127) AND
      C$ < > CHR$ (B) THEN N$
      = N$ + C$: PRINT C$;: G
      OTO 1320
CE 1340 IF N$ = "" THEN 1320
29 1350 HTAB LEN (N$) + 14: PRIN
      T " ";: HTAB LEN (N$) +
      14: IF LEN (N$) = 1 THEN
      N$ = "": GOTO 1320
BA 1360 N$ = LEFT$ (N$, LEN (N$)
      - 1): GOTO 1320
0C 1370 F$ = "LOAD": GOSUB 1270:
      IF LEN (N$) < > 0 THEN
      PRINT : PRINT O$;"BLOAD
      ";N$;",A$B4C4"
01 13B0 GOTO 590
EA 1390 PRINT : PRINT "AN ERROR
      HAS OCCURRED"
5E 1400 PRINT "MAKE SURE THAT YO
      U HAVE A FORMATTED"
F9 1410 PRINT " DISK IN THE DRIV
      E"
49 1420 PRINT : PRINT "PRESS ANY
      KEY TO CONTINUE"
C2 1430 GET W$: GOSUB 590: GOTO
      170
BE 1440 GOSUB 700: VTAB 19: HTAB
      10: PRINT "QUIT ANIMATO
      R";: GOSUB 1160: IF C =
      217 THEN MQ = 1: RETURN
F6 1450 GOSUB 660: RETURN
A7 1460 CALL 32736: GOTO 520
BD 1470 GOSUB 700: VTAB 19: HTAB
      9: PRINT "CLEAR ALL BOX
      ES";: GOSUB 1160: IF C =
      206 THEN 1490
78 14B0 GOSUB 700: GOSUB 650: GO
      SUB 520
07 1490 GOSUB 660: RETURN
36 1500 AB = F: RETURN
9A 1510 AE = F: RETURN
40 1520 A = AB:QF = 0:AR = 0:AX
      = 0
BD 1530 CALL 32771,A0,60: CALL 3
      276B,A,AX,60:A0 = AX: IF
      QF = 1 THEN RETURN
0B 1540 C = PEEK (49152): IF C >
      12B THEN POKE 4916B,0:
      GOSUB 1600
EX 1550 FOR I = 0 TO AP * 5: NEX
      T :AR = AR + AS: IF AR >
      259 THEN AR = 0
5C 1560 IF AR < 0 THEN AR = 259
BA 1570 AX = 2 * INT (AR / 2): I
      F AE > AB THEN A = A + 1
      : IF A > AE THEN A = AB
3B 15B0 IF AE < AB THEN A = A -
      1: IF A < AE THEN A = AB
BE 1590 GOTO 1530
61 1600 IF C < > 160 THEN 1630
9A 1610 IF PEEK (49152) < 12B TH
      EN 1610
03 1620 POKE 4916B,0: RETURN
36 1630 IF C = 136 THEN GOSUB 16
      70: GOTO 710
0B 1640 IF C = 149 THEN GOSUB 16
      60: GOTO 710
43 1650 QF = 1: RETURN
1B 1660 AS = AS + (AS < 15): RET
      URN
D2 1670 AS = AS - (AS > - 15): R
      ETURN
5A 16B0 AP = AP - (AP > 0): RETU
      RN
01 1690 AP = AP + (AP < 150): RE
      TURN
BC 1700 DATA QUIT,DRAW,MOVE,ERAS
      E,CLEAR,SAVE,INVERT,UPDA
      TE,REVERT
9B 1710 DATA BLACK1,GREEN,PURPLE
      ,WHITE1,BLACK2,ORANGE,BL
      UE,WHITE2
B0 1720 DATA EDIT FRAME,INSERT F
      RAME,DELETE FRAME,RANGE
      BOTTOM,RANGE TOP
E6 1730 DATA ANIMATE,FASTER SPEE
      D,SLOWER SPEED,LESS PAUS
      E
A0 1740 DATA MORE PAUSE,LOAD,SAV
      E,CLEAR,INVERT,QUIT
```

Program 7: TI Animator

Version by Patrick Parrish, Programming Supervisor

```
10 FORE=2 :: BACK=12 :: REM
    REQUIRES EXTENDED BASIC
20 CALL CLEAR :: GOSUB 4B0
   :: F=1 :: L=10 :: N=1 ::
   BE=1 :: E=20 :: SP=0 ::
   OL=0 :: FOR I=104 TO 14
   3 :: CALL CHAR(I,RPT$("0
   ",16)):: NEXT I
30 CALL MAGNIFY(4):: DIM B(
   16,16),C$(15),E$(20),IFL
   AG(20):: Q$="OEVICE (OSK
```

```
   1.FILE OR CS1)?" :: GOSU
   B 910 :: CALL SCREEN(BAC
   K):: GOSUB 970
40 CALL KEY(0,K,S):: IF S=0
   THEN 40
50 IF K>4B AND K<57 THEN K=
   K-4B :: ON K GOTO 170,50
   0,220,220,240,240,250,25
   0
60 IF K=66 THEN GOSUB 260
70 IF K=70 THEN GOSUB 270
80 IF K=73 THEN GOSUB 2B0
90 IF K=69 THEN GOSUB 300
100 IF K=65 THEN GOSUB 330
110 IF K=76 THEN GOSUB 370
120 IF K=67 THEN GOSUB 390
130 IF K=81 THEN END
140 IF K=7B THEN GOSUB 410
150 IF K=6B OR K=B3 THEN N=
   N-((N<>1)-(N=1)*19)*(K=
   B3)+((N<>20)-(N=20)*19)
   *(K=6B):: OISPLAY AT(4,
   1):N;
160 GOTO 40
170 CALL CLEAR :: GOSUB 4B0
    :: FOR I=5 TO B :: CAL
   L COLOR(I,FORE,1):: NEX
   T I :: FOR I=1 TO 20 ::
    CALL CHAR(60+I*4,E$(I)
   ):: NEXT I
1B0 GOSUB 490 :: CALL SPRIT
   E(#1,60+BE*4,FORE,100,1
   00):: CALL MOTION(#1,0,
   SP*6)
190 FOR I=BE TO E :: CALL P
   ATTERN(#1,60+I*4):: OIS
   PLAY AT(1,1):I :: FOR J
   =1 TO OL :: NEXT J :: N
   EXT I
200 CALL KEY(0,K,S):: IF S=
   0 THEN 190 ELSE CALL OE
   LSPRITE(#1):: CALL CHAR
   SET
210 FORE=FORE-1 :: GOSUB 27
   0 :: FOR I=5 TO B :: CA
   LL COLOR(I,2,1):: NEXT
   I :: LF=1 :: GOSUB 910
   :: GOSUB 970 :: GOTO 40
220 IF K=3 THEN BE=N ELSE E
   =N
230 GOSUB 1060 :: GOTO 40
240 SP=SP-(SP<>-15)*(K=5)+(
   SP<>15)*(K=6):: GOSUB 1
   060 :: GOTO 40
250 OL=OL-(OL<>0)*(K=7)+(OL
   =0)*(K=7)*150+(OL<>150)
   *(K=B)-(OL=150)*(K=B)*1
   50 :: GOSUB 1060 :: GOT
   O 40
260 BACK=BACK+1+(BACK=16)*1
   4 :: CALL SCREEN(BACK)::
    RETURN
270 FORE=FORE+1+(FORE=16)*1
   5 :: FOR I=10 TO 14 ::
   CALL COLOR(I,FORE,1)::
   NEXT I :: CALL COLOR(9,
   FORE,16):: RETURN
2B0 GOSUB 320 :: IF A$="N"
   THEN RETURN
290 FOR I=20 TO N+1 STEP -1
    :: E$(I)=E$(I-1):: NEX
   T I :: E$(N)=RPT$("0",6
   4):: GOSUB 450 :: GOSUB
    440 :: RETURN
300 GOSUB 320 :: IF A$="N"
   THEN RETURN
310 FOR I=N TO 19 :: E$(I)=
   E$(I+1):: NEXT I :: E$(
   20)=RPT$("0",64):: GOSU
   B 450 :: GOSUB 440 :: R
   ETURN
320 OISPLAY AT(12,4):"ARE Y
   OU SURE (Y/N)?" :: ACCE
   PT AT(12,24)VALIDATE("Y
```



"TI Animator" makes extensive use of redefined character graphics.

```
   N"):A$ :: IF A$="Y" THE
   N RETURN ELSE GOSUB 440
    :: RETURN
330 GOSUB 420 :: OPEN #1:O$
   ,INTERNAL,OUTPUT,FIXED
   80 :: PRINT #1:FORE ::
   PRINT #1:BACK :: FOR I=
   1 TO 20 :: PRINT #1:E$(
   I)
340 PRINT #1:IFLAG(I):: NEX
   T I
350 CLOSE #1 :: FOR I=10 TO
    14 :: CALL COLOR(I,FOR
   E,1):: NEXT I :: CALL S
   CREEN(BACK)
360 IF ASC(O$)=67 THEN GOSU
   B 970 :: RETURN ELSE GO
   SUB 440 :: RETURN
370 GOSUB 420 :: OPEN #1:O$
   ,INTERNAL,INPUT ,FIXED
   80 :: INPUT #1:FORE ::
   INPUT #1:BACK :: FOR I=
   1 TO 20 :: INPUT #1:E$(
   I)
3B0 INPUT #1:IFLAG(I):: NEX
   T I :: GOSUB 450 :: GOT
   O 350
390 GOSUB 320 :: IF A$="N"
   THEN RETURN
400 FOR I=1 TO 20 :: E$(I)=
   RPT$("0",64):: NEXT I :
   : GOSUB 450 :: GOSUB 44
   0 :: RETURN
410 F=F-(F=1)*10+(F=11)*10
   :: L=F+9 :: GOSUB 450 :
   : GOSUB 460 :: RETURN
420 OISPLAY AT(12,1):O$ ::
   ACCEPT AT(13,1):O$ :: I
   F O$="" THEN GOSUB 440
430 RETURN
440 CALL HCHAR(12,1,32,64):
   : RETURN
450 J=0 :: FOR I=F TO L ::
   CALL CHAR(104+J*4,E$(I)
   ):: J=J+1 :: NEXT I ::
   RETURN
460 CALL HCHAR(9,2,32,2)::
   J=F+1 :: FOR I=2 TO 26
   STEP 3 :: OISPLAY AT(9,
   I):J :: J=J+1 :: NEXT I
    :: F$=STR$(F):: OISPLA
   Y AT(4,1):N;
470 FOR I=1 TO LEN(F$):: CA
   LL HCHAR(9,1+I,ASC(SEG$
   (F$,I,1))):: NEXT I ::
   RETURN
4B0 OISPLAY AT(1,7):"PLEASE
    WAIT..." :: RETURN
490 CALL HCHAR(1,9,32,14)::
    RETURN
500 OISPLAY AT(12,1):"TYPE
   21 TO ABORT, A # 1-20,
   OR <ENTER> FOR CURRENT
   # :"
510 ACCEPT AT(13,27):A$ ::
   IF A$="" THEN PN=N :: G
   OTO 560
520 IF A$<"1" OR A$>"9" THE
   N 510
530 J=VAL(A$):: IF J=21 THE
   N GOSUB 440 :: GOTO 40
540 IF J>20 OR J<1 THEN 510
550 IF J=N THEN PN=N ELSE P
   N=J
560 IFLAG(PN)=IFLAG(N):: CA
   LL CLEAR :: OISPLAY AT(
   2,1):"1-MOVE  2-ORAW
   (4 SPACES)3-ERASE" :: O
   ISPLAY AT(3,1):"4-CLEAR
    5-INVERSE 6-OISPLAY"
570 OISPLAY AT(4,1):"7-SAVE
     B-QUIT" :: OISPLAY AT
   (5,2):"(USE ARROW KEYS
   TO MOVE)" :: OISPLAY AT
   (24,7):"FRAME";PN :: GF
   =1 :: GOSUB 920
5B0 IF IFLAG(N)=1 THEN CALL
    CHAR(100,U$,101,"")ELS
   E CALL CHAR(100,"",101,
   U$)
590 F$=E$(N):: FOR I=0 TO 6
   3 STEP 2 :: O=ASC(SEG$(
   F$,I+1,1))-4B :: O=O+(O
   >9)*7 :: O1=ASC(SEG$(F$
   ,I+2,1))-4B :: O1=O1+(O
   1>9)*7
600 O=-15*(IFLAG(PN)=1)+O*(
   IFLAG(PN)=1)-(IFLAG(PN)
   =0)*O :: O1=-15*(IFLAG(
   PN)=1)+O1*(IFLAG(PN)=1)
   -(IFLAG(PN)=0)*O1
610 OISPLAY AT(7+I/2+(I>31)
   *16,3-(I>31)*B):C$(O):: 
    OISPLAY AT(7+I/2+(I>31
   )*16,7-(I>31)*B):C$(O1)
   :: NEXT I
620 CALL SPRITE(#28,96,14,4
   9,33):: KHAR=101 :: FLA
   G=1 :: R=1 :: C=1
630 CALL KEY(0,K,S):: IF S=
   0 THEN 630 ELSE CALL OE
   LSPRITE(#1)
640 IF K=51 THEN KHAR=100 :
   : FLAG=0 :: GOTO 790
650 IF K=49 THEN FLAG=1 ::
   GOTO B00
660 IF K=50 THEN KHAR=101 :
   : FLAG=0 :: GOTO 790
670 IF K<>52 THEN 690
6B0 FOR R=1 TO 16 :: CALL H
   CHAR(6+R,5,100,16):: NE
   XT R :: OFLAG=0 :: GOTO
    620
690 IF K<>53 THEN 720
700 OFLAG=0 :: IFLAG(PN)=-(
   IFLAG(PN)=0):: IF IFLAG
   (PN)=0 THEN CALL CHAR(1
   00,"",101,U$)ELSE CALL
   CHAR(100,U$,101,"")
710 GOTO B00
720 IF K=56 THEN CALL OELSP
   RITE(ALL):: LF=1 :: GOS
   UB 910 :: GOSUB 970 ::
   GOTO 40
730 IF K=66 THEN GOSUB 260
   :: OFLAG=1
740 IF K=70 THEN GOSUB 270
   :: OFLAG=1
750 IF K=54 OR K=55 THEN CA
   LL OELSPRITE(ALL):: IF
   OFLAG=1 THEN B90 ELSE B
   10
760 C=C+(K=B3)+((C=1)*(K=B3
   )*16)-(K=6B)-((C=16)*(K
   =6B)*16)
770 R=R+(K=69)+((R=1)*(K=69
   )*16)-(K=BB)-((R=16)*(K
   =BB)*16)
780 CALL LOCATE(#2B,B*R+41,
```

```
    8*C+25):: IF FLAG=1 THE
    N 800
790 OFLAG=0 :: CALL HCHAR(6
    +R,4+C,KHAR)
800 CALL SOUND(20,200,5):: 
    GOTO 630
810 GOSUB 480 :: FOR R=1 TO
     16 :: FOR C=1 TO 16 ::
     CALL GCHAR(6+R,4+C,GC)
    :: GC=GC-100 :: B(R,C)=
    GC :: NEXT C :: NEXT R
820 F$="" :: IF IFLAG(PN)=0
     THEN HEX$="0123456789A
    BCDEF" ELSE HEX$="FEDCB
    A9876543210"
830 FOR R=1 TO 16 :: LDW=B(
    R,5)*8+B(R,6)*4+B(R,7)*
    2+B(R,8)+1
840 HIGH=B(R,1)*8+B(R,2)*4+
    B(R,3)*2+B(R,4)+1
850 F$=F$&SEG$(HEX$,HIGH,1)
    &SEG$(HEX$,LOW,1):: NEX
    T R :: FOR R=1 TO 16
860 LOW=B(R,13)*8+B(R,14)*4
    +B(R,15)*2+B(R,16)+1
870 HIGH=B(R,9)*8+B(R,10)*4
    +B(R,11)*2+B(R,12)+1
880 F$=F$&SEG$(HEX$,HIGH,1)
    &SEG$(HEX$,LOW,1):: NEX
    T R :: IF K=54 THEN DFL
    AG=1
890 IF K=55 THEN E$(PN)=F$
    :: LF=1 :: GOSUB 910 ::
     GOSUB 970 :: GOTO 40
900 CALL MAGNIFY(4):: CALL
    CHAR(36,F$):: GOSUB 490
     :: CALL SPRITE(#1,36,F
    ORE,80,175):: GOTO 620
910 CALL CHAR(33,RPT$("81",
    8),36,RPT$("0",14)&"FF"
    ,37,"FF"&RPT$("0",14),3
    8,RPT$("01",8),39,RPT$(
    "80",8)):: IF LF=1 THEN
     LF=0 :: RETURN
920 U$=RPT$("F",16):: CALL
    CHAR(100,"",101,U$,96,"
    F09090F"&RPT$("0",57)):
    : CALL COLOR(9,FORE,16)
    :: IF GF=1 THEN GF=0 ::
     RETURN
930 F$="0000000010010000110
    0010101100111100010011
    0
    1010111100110111101111"
940 FOR I=0 TO 15 :: Z$=SEG
    $(F$,I*4+1,4):: O$=""
950 FOR J=1 TO 4 :: T=VAL(S
    EG$(Z$,J,1))+100 :: O$=
    O$&CHR$(T):: NEXT J ::
    C$(I)=O$ :: NEXT I
960 FOR I=1 TO 20 :: E$(I)=
    RPT$("0",64):: NEXT I :
    : RETURN
970 CALL CLEAR :: DISPLAY A
    T(1,10):"ANIMATOR" :: C
    ALL HCHAR(4,2,35):: DIS
    PLAY AT(4,5):"B=
    {4 SPACES}E={4 SPACES}S
    P={4 SPACES}D="
980 GOSUB 1060 :: CALL VCHA
    R(6,1,38,2):: CALL VCHA
    R(6,31,39,2):: FOR I=2
    TO 29 STEP 3 :: CALL HC
    HAR(5,I,36,2):: CALL HC
    HAR(8,I,37,2)
990 NEXT I :: FOR I=4 TO 28
     STEP 3 :: CALL VCHAR(6
    ,I,33,2):: NEXT I :: GO
    SUB 460 :: GOSUB 450 ::
     J=104
1000 FOR I=2 TO 29 STEP 3 :
    : CALL HCHAR(6,I,J):: 
    CALL HCHAR(6,I+1,J+2):
    : CALL HCHAR(7,I,J+1):
    : CALL HCHAR(7,I+1,J+3
    ):: J=J+4 :: NEXT I
1010 CALL HCHAR(15,1,49):: 
    CALL HCHAR(16,1,52):: 
    CALL HCHAR(17,1,55)
1020 DISPLAY AT(15,1):"ANIM
    ATE 2 EDIT{3 SPACES}3
    START PIC" :: DISPLAY
    AT(16,1):"END PIC 5 -S
    PEED 6 +SPEED"
1030 DISPLAY AT(17,1):"-SWI
    TCH 8 +SWITCH"
1040 DISPLAY AT(20,1):"(I)N
    SERT PIC  O(E)LETE PIC
    " :: DISPLAY AT(21,1):
    "S(A)VE  (L)OAD  (C)LE
    AR ALL"
1050 DISPLAY AT(22,1):"(N)E
    XT GROUP{6 SPACES}(Q)U
    IT" :: RETURN
1060 DISPLAY AT(4,7):BE;:: 
    DISPLAY AT(4,13):E;:: 
    DISPLAY AT(4,20):SP;:: 
     DISPLAY AT(4,26):STR$
    (DL)&SEG$("{3 SPACES}"
    ,1,3-LEN(STR$(DL)));:: 
     RETURN
```

©

Archive:
Two-Drive Backup for Commodore 64

Philip I. Nelson, Assistant Editor

Now you can copy entire disks at machine language speed with this convenient backup program for the Commodore 64 with two 1541 disk drives. It also works on the new Commodore 128 in 64 mode.

Sooner or later it's bound to happen. You'll make an unconscious error, or lightning may strike while you're resaving a program, or the family dog will chew a few disks for dinner—and an important disk will be utterly destroyed. If you have a backup copy, of course, such accidents aren't fatal. You take a moment to pat yourself on the back, pull out the archive disk, and go back to work.

If you don't have a backup, it's like watching a gold ring slip off your finger and go clanking down the drain. In the long hours spent reconstructing what you've lost, you have plenty of time to reflect on the wisdom of archiving your work on a regular basis.

Archiving is one of those grim tasks that's easy to postpone. BASIC programs (like COPY/ALL on your 1541 Test/Demo disk) are slow, and may not copy machine language (ML) programs or sequential files. Even good single-drive backup programs keep you tied to the computer, tapping your fingers until it's time for the next disk swap.

Fast And Convenient

"Archive" offers a better way: It links two drives together to take the misery out of backing up important disks. To speed things up, it's written entirely in machine language and copies only those disk sectors which actually contain data. But because it loads and runs just like a BASIC program, it's easy for anyone to use, even beginners.

You may find this program valuable even if you don't own two disk drives. Put your drive together with a friend's and swap several disks during one session. Or bring it to a user group meeting to speed up the duplication of public domain library disks. Since 1541-format disks work with other Commodore computers, Archive running on a Commodore 64 can also copy disks that will be used with the Commodore 128, VIC-20, Plus/4, 16, and 4040-format PET/CBM. (Of course, a program written for one of these machines may not work on another. Also, Archive cannot copy Commodore 128 CP/M disks.)

Archive has been tested successfully on the Commodore 128 in 64 mode with two 1541 drives. If the new 1571 drives are truly 1541-compatible, Archive will work with them as well, since it uses standard Commodore disk commands. However, the 1571 was not available for testing when this article was written.

Incidentally, Archive cannot duplicate commercially protected software. Protected disks invariably contain deliberate errors (which shut down the program) or data hidden in unused sectors (which Archive does not copy).

Getting Started

Enter and save Archive using the MLX machine language entry program published elsewhere in this issue. Here's the information you need:

Starting address: 49152
Ending address: 51185

After you save Archive, activate it like a BASIC program by typing LOAD"ARCHIVE",8 followed by RUN. (For this program, *do not* use ,8,1 after the LOAD; just use ,8.) If you're already comfortable using two drives, you needn't read any further, since Archive prompts you at each step. Just pop a disk in each drive as instructed, press the f7 special function key, and relax while Archive does its work. (If you've never used two drives before, see "Setting Up Your System" below.)

Archive displays your source disk's Block Availability Map (BAM) graphically on the screen, updating the display as copying proceeds. Thus, you can tell at a glance how much of the disk is used and how much has been copied. The number at the lower right of the screen shows the sector being copied; the graphic display shows which sectors have already been copied.

If you want to abort the copy for any reason, press the f1 special function key to return to BASIC. (When you abort the copy process, the archive disk is incomplete and may be garbled. You can reuse it immediately with Archive, but do not use it for anything else without reformatting it as explained below.)

Once the copy is done, press the f3 function key to copy another disk, or press f1 to quit. Whenever you exit Archive, it clears the screen and reports the status of each drive.

Quick Formatting

Since Archive always makes a complete disk copy, it *formats* the archive disk with a NEW command. Formatting renames the disk and erases everything it contained before. You'll notice that the archive disk is formatted in only a few seconds rather than the usual couple of minutes, and without the usual knocking sound. To save time and minimize wear on the drive, Archive uses a shortened NEW command: the equivalent of OPEN 15,8,15,"N0:*filename*" without a disk ID.

The abbreviated NEW command works only on a disk that has been previously formatted. To use a brand new disk, you must prepare it first with a full NEW command: OPEN 15,8,15,"N0:*filename,ID*". The filename can be up to 16 characters long. The ID can be any two letters or numbers and should be unique for each disk. The *1541 User's Manual* contains more information about formatting disks.

Setting Up Your System

Although the Commodore 64 has only one serial port connector, the 1541 disk drive has two, letting you hook up more than one drive at a time. Since the drives are chained together in a series, this arrangement is often called *daisy-chaining*. Figure 1 shows how to daisy-chain two drives to your computer. Connect the first drive as usual, then plug the serial cable from the second drive into the extra serial port connector on the first drive.

When more than one disk drive is active, each drive must be given a different *device number* so the computer can tell them apart. The 1541 is factory-set as device 8, but it can also have device numbers 9–15. Archive uses device numbers 8 and 9, reading from drive 8 and copying to drive 9. *You must always put the source disk (the original) in the drive that's device 8 and the archive disk (the copy) in the drive that's device 9.*

If both of your drives are device 8, don't despair. You can easily change one of them to device 9. The change is temporary; the drive reverts to device 8 when you turn off the power. Here's the procedure:

1. Turn on the drive that you want to change to device 9. *Make sure the other drive is turned off.* Now you can change the device number either by running the DISK ADDR CHANGE utility program on your 1541 Test/Demo disk, or by typing in direct statements.

To use DISK ADDR CHANGE, load the program from the 1541 Test/Demo disk and enter RUN. Follow the program's instructions, then skip to Step 2 below.

You can also change the device number by entering the following statements in direct mode (with no line numbers). Press RETURN after you type each line:

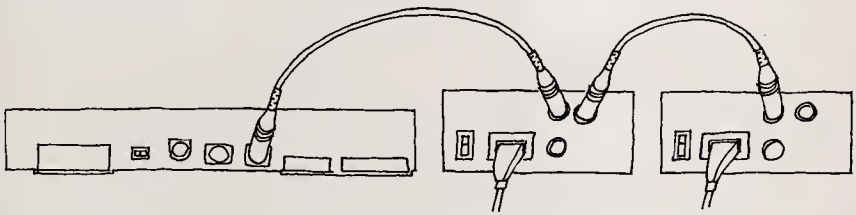
```
OPEN 15,8,15
PRINT#15,"M-W"CHR$(119)
  CHR$(0)CHR$(2)CHR$(32+9)
  CHR$(64+9)
CLOSE15
```

2. It's a good idea to verify the device number change. Put a disk in the drive, then type LOAD"$",9 and press RETURN to load its directory. After the blinking cursor returns, type LIST and press RETURN. If you see the directory, the change worked and you may proceed to step 3. If you get an error (probably ?DEVICE NOT PRESENT), turn off the drive and repeat step 1.

3. Turn on the other drive. This drive will remain device 8 (the source drive). Now load and run Archive, inserting the disks as explained in the instructions. The source (original) disk goes in device 8, and the archive (copy) disk goes in device 9. As an additional precaution, you may want to write-protect the source disk by taping over the notch in the sleeve.

In theory you can daisy-chain several drives to a 64, but in fact the 1541 doesn't enjoy sharing the serial bus. The drives should always be turned on one at a time, not simultaneously (as would happen with a power strip). Printer interfaces that draw power from the 64's cassette port are notorious for causing disk errors, and other peripherals can affect system voltage levels even if they're not turned on. Depending on your system, you may need to unplug other peripherals before using Archive.

Daisy-Chaining Two Disk Drives



The second drive is plugged into the first drive's extra serial port connector (rear view).

Trusted Software and

LANGUAGE SOFTWARE

ADA TRAINING COURSE

Teaches you the language of the future. Comprehensive subset of language. Includes: editor; syntax checker, compiler; assembler; disassembler, handbook. $59.95

ASSEMBLER/MONITOR-64

Macro assembler and extended monitor. Supports all standard functions plus floating point constants Monitor supports bank switching, quick trace, single step, more. $39.95

BASIC COMPILER 64

Compiles *complete* BASIC language into either fast 6510 machine language and/or compact speedcode. Supports overlays and many extended BASIC commands. $39.95

C LANGUAGE COMPILER

Full compiler per Kernighan & Ritchie standard, but without bit fields. Includes powerful editor (41K source file); compiler (6510 code), library (supports many functions) and linker. NEW! $79.95

FORTH-64 LANGUAGE

Based on Forth 79 (+ parts of '83). Supports hires graphics and sound synthesizer. Full screen editor, programming tools, assembler, samples, handbook. NEW! $39.95

MASTER-64

Professional development package for serious applications. Indexed file system, full screen & printer management, programmer's aid, multiprecision math, royalty-free runtime, more. $39.95

PASCAL-64

Full Pascal supports graphics, sprites, file management, more. Supports pointers, dynamic memory management, machine language. Compiles to fast 6510 machine code. $39.95

VIDEO BASIC-64

Add 50+ graphic, sound and utility commands to your programs with this super development package. Free distribution of RUNTIME version - no royalties! $59.95

XREF-64 Basic cross-reference

Indispensible tool for BASIC programmer. Cross-references all references to variables, constants and line numbers. Sorts them into alphabetical order for convenient uses. $17.95

FOR COMMODORE COMPUTERS

REFERENCE BOOKS

ANATOMY OF C-64 Insider's guide to the '64 internals. Graphics, sound, I/O, kernal, memory maps, more. Complete commented ROM listings. 300pp $19.95

TRICKS & TIPS FOR C-64 Collection of easy-to-use techniques: advanced graphics, improved data input, enhanced BASIC, CP/M, more. 275pp $19.95

ANATOMY OF 1541 DRIVE Best handbook on floppy explains all. Many examples and utilities. Fully commented 1541 ROM listings. 320pp $19.95

1541 REPAIR & MAINTENANCE Handbook describes the disk drive hardware. Includes schematics and techniques to keep 1541 running. 200pp $19.95

MACHINE LANGUAGE C-64 Learn 6510 code write fast programs. Many samples & listings for complete assembler, monitor, simulator. 200pp $14.95

ADVANCED MACHINE LANGUAGE Not covered elsewhere: - video controller, interrupts, timers, clocks, I/O, real time, extended BASIC, more. 210pp $14.95

GRAPHICS BOOK C-64 - best reference covers basic and advanced graphics. Sprites, animation, Hires, Multicolor, lightpen, 3D-graphics, IRQ, CAD, projections, curves,more. 350pp $19.95

PRINTER BOOK C-64/VIC-20 Understand Commodore; Epson-compatible printers; 1520 plotter. *Packed:* utilities; graphics dump; 3D-plott; commented MPS801 ROM listings, more. 330pp $19.95

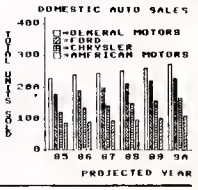
PRODUCTIVITY TOOLS

CHARTPAK-64

MAKE YOUR OWN CHARTS...

Professional quality charts and graphs instantly. Easy-to-use menus. 8 chart formats. Two size hardcopy. Accepts data from Multiplan, CalcResult. Supports statistical functions. $39.95

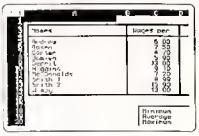
Also available:
CHARTPLOT-1520 fine output to inexpensive 1520 plotter. $39.95
CHARTPLOT-64 unsurpassed quality charts on most XY-plotters. $84.95



POWER PLAN-64

Super Spreadsheet with Graphs

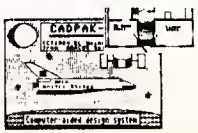
POWER PLAN is our top-rated, full-featured spreadsheet for both your business and personal needs. All menu selections are clearly displayed on screen. Built-in HELP screens guide you through this powerful package. Immediately make 2D or 3D pie, bar, line and point to point graphs from your spreadsheet using up to 8 windows. Features cell protection and formatting, split screen, all major math functions, 25K worksheet space, 255 rows X 63 columns. 200 page handbook. $49.95



CADPAK-64

DETAIL YOUR DESIGNS....

Superb lightpen design tool. Exact placement using *Accu-Point* positioning; 2 complete screens; draw LINEs, BOXes, CIRCLEs, ELLIPSEs; pattern FILLing; freehand DRAW; screen COPY; ZOOM for detail. Two size hard-copy. Requires good quality lightpen. $49.95



Authoritative books

FROM ABACUS SOFTWARE ... A NAME YOU CAN COUNT ON

Optional Diskettes for books

For your convenience, the programs contained in each of our books are available on diskette to save you time entering them from your keyboard. Specify name of book when ordering. **$14.95 each**

SCIENCE/ENGINEERING ON C-64 In depth intro to computers in science. Topics: chemistry, physics, biology, astronomy, electronics, others. 350pp $19.95

CASSETTE BOOK C-64/VIC-20 Comprehensive guide; many sample programs. High speed operating system-fast file loading and saving. 225pp $14.95

IDEAS FOR USE ON C-64 Themes: auto expenses, calculator, recipe file, stock lists, diet planner, window advertising, others. Includes listings. 200pp $12.95

COMPILER BOOK C-64/C-128 All you need to know about compilers: how they work; designing and writing your own; generating machine code. With working example compiler. 300pp $19.95

PEEKS & POKES - Programming quickies that will amaze you. Includes in-depth explanations of PEEK, POKE, USR and other commands. 200pp $14.95

ADVENTURE GAMEWRITER's Handbook Step-by-step guide to designing and writing your own adventure game generator. 200pp $14.95

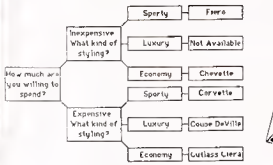
CAD BOOK for C-64/C-128 Introduction to computer aided design. 2D and 3D, reflection, zooming, macros. Samples use SIMON's Basic. 250pp $19.95

MORE TRICKS & TIPS Dozens of more helpful programming techniques: software protection; new BASIC commands; interrupts; ROM routines; the KERNAL, hardware tips. etc. 250pp $19.95

Other Books Coming Soon!!

SPECIAL FEATURE

XPER - Software that Learns



An XPER Application advising on what kind of car to buy

XPER is the first *expert system* for the C-64 and C-128. Ordinary data bases are good for reproducing facts, but XPER lets you derive knowledge from a mountain of facts and helps you make expert decisions. Scientists, doctors and professionals are currently using XPER in their research projects.

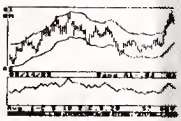
The XPER system puts your information into a *knowledge base*. A knowledge base consists of a variable number of **objects** - such as trees; **features** - such as "what kind of leaf does it have?"; and **attributes** - such as 6 inch needles. After describing your knowledge base, you can easily enter your information using XPER's complete entry/maintenance facilities.

Then, XPER's very efficient searching techniques will guide you through the most complex decision making criteria. You'll also benefit from XPER's full reporting capabilities and ease of use. **$59.95**

TAS-64

CHART YOUR OWN STOCKS...

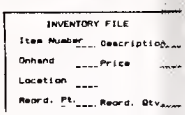
Sophisticated technical analysis charting package for serious investor. Capture data from DJN/RS or Warner Services or enter and edit your data. 7 moving averages, 3 oscillators, trading bands, least squares, 5 volume indicators, relative charts, much more. Two size hardcopy. $84.95



DATAMAT-64

ORGANIZE YOUR DATA...

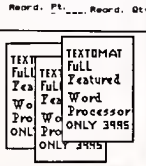
Powerful, easy-to-use data management package uses menu selections. Free form design: 50 fields/record; 2000 records/disk. Sort on multiple fields in any combination. Complete selection and formatting for printing reports. $39.95



TEXTOMAT-64

WORD PROCESSING.....

Flexible wordprocessor displays 40 or 80 columns with horizontal scrolling. Menu selections make TEXTOMAT easy to use. Quickly move from editing to formatting to merging to utilities. Supports virtually any printer. Can support foreign languages. $39.95



QUICKCOPY V2.0

NEW!

PROTECT Your DATA ...

Backup your valuable data with the fastest disk copier we've seen to date. Copies entire disk in 2.5 minutes on two or 3.5 minutes on one 1541. Very useful utility. $19.95

ORDERING INFORMATION

Call now for the name of your nearest dealer

Abacus Software

P.O. Box 7211 Grand Rapids, Michigan 49510

MasterCard VISA AMERICAN EXPRESS Card

For postage and handling include $4.00 per order. Foreign orders include $8.00 per *item*. Money order and checks in U.S. Dollars only. Mastercard, VISA and American Express accepted. Michigan residents please include 4% sales tax.

For fast service call (616) 241-5510 Telex 709-101

For **free** catalog, please return this coupon to

Abacus Software, P.O. Box 7211, Grand Rapids, MI 49510

PHONE: (616) 241-5510

Name _____

Address _____

City _____

State _____ **Zip** _____

Compute 8/85

Initialization

Before it starts copying, Archive *initializes* each disk to test whether devices 8 and 9 are active and if each contains a formatted disk. The initialization command transfers information (disk name, ID, etc.) from the disk into the drive's memory to prepare it for handling the disk.

If this step fails, it means one of the drives is not ready to go. Archive displays the status of both drives and returns you to BASIC. (If you forget to change one of the drives to device 9, Archive can't read its status; press RUN/STOP –RESTORE and proceed as explained below). Enter these lines in direct mode (without a line number) to retry the initialization:

```
OPEN 15,8,15,"I0":CLOSE 15
OPEN 15,9,15,"I0":CLOSE 15
```

When you enter each line, the drive motor should run and its red light should glow. After one or two seconds the red light should go off and stay off, and you should be able to run Archive. If one or both of the red lights blink continuously, turn off both drives and repeat the setup process. The drive makes a knocking sound if you forget to insert a disk or try to use an unformatted disk for the archive.

Is Your Drive Healthy?

In ordinary use your drive works intermittently. It may spend 30 seconds loading a game for you, then sit idle for 30 minutes while you play. Copying a full disk with Archive is far more demanding work, requiring several minutes of continuous running. If one of your drives is misaligned, tends to overheat, or has other mechanical problems, don't be surprised if you experience occasional errors. When an error occurs during the copy process, Archive stops copying, reports the status of both drives, and returns to BASIC.

Such errors are especially likely to crop up when the source disk is nearly full. For mechanical reasons it's harder for the drive to access the disk's outer area than the area near the middle. To make things easy on itself, the drive always starts storing programs in the middle of the disk, leaving the outer tracks empty until there's no room left elsewhere.

Archive's BAM display lets you observe this storage scheme. When the source disk contains only a few programs, they'll all be stored in middle tracks (near track 18). The outermost tracks (1 and 35) are usually the last to be used. If your drive consistently has trouble accessing outer tracks, it's probably misaligned. The same problem can result if the disk was formatted on a badly misaligned drive.

1541 ML Programming

To shorten and speed up the program, all of Archive's 21 variables and pointers are located in the zero page (lowest 256 bytes) of memory. Zero page machine language instructions run faster and use less memory than instructions that reference higher memory addresses. The computer can find what it needs by checking only one byte, rather than wading through a two-byte address in search of the same information. In time-critical programs like Archive, which execute certain routines many thousands of times a minute, the microseconds you save can add up to a significant difference in running time.

Many programmers have trouble learning to handle disk files in machine language. For those who are interested, here's an outline of Archive's main routines.

$0852–0863	Initialize device 8
$0864–0878	Initialize device 9
$0879–0918	Error—report status
$0919–0981	Read BAM from source disk
$0982–0A32	Display BAM and disk name
$0A33–0A7E	Short NEW destination disk
$0A7F–0A8F	OPEN 3,8,15 command channel
$0A90–0AA0	OPEN 5,9,15 command channel
$0AA1–0A84	OPEN 4,8,4,"#" buffer channel
$0A85–0AC8	OPEN 6,9,6,"#" buffer channel
$0894–0C42	Subroutine—copy a block
$00CE–0C04	Read block from source disk
$0C05–0C42	Write block to archive disk
$0C43–0C5C	Subroutine—initialize disk
$0C94–0CA4	Subroutine—check error channel
$0CC0–0CC8	String—BAM Block Read (U1)
$0CCC–0CDF	String buffer—short NEW
$0CE0	String—"#" for buffer channel
$0CE1–0CE2	String—"I0" to initialize
$0F04–0F0F	String buffer—Block Read (U1)
$0F10–0F1Ø	String—Block Write (U2)
$0FF3	256-byte data buffer starts here

First the program maps the source disk's BAM on the screen to record which sectors contain data. Then it copies each used sector in turn, reading it from the source disk and writing it to the archive disk. Note that to read a disk sector, you should always use the U1 direct access command rather than B-R (Block Read). Likewise, the U2 command must be used in place of B-W (Block Write). Despite what your user's guide says, the B-R and B-W commands are defective and should never be used.

Archive: Two-Drive Backup

Please refer to the "MLX" article in this issue before entering the following listing.

```
49152 :027,008,000,000,158,050,243
49158 :048,055,055,058,143,034,143
49164 :020,020,020,020,020,032,144
49170 :065,082,067,072,073,086,207
49176 :069,000,000,000,162,015,014
49182 :142,033,208,142,032,208,027
49188 :232,142,134,002,169,028,231
49194 :160,015,032,030,171,169,107
49200 :124,133,176,133,178,169,193
49206 :004,133,177,169,216,133,118
49212 :179,165,197,201,004,240,022
49218 :006,201,003,208,246,240,202
49224 :008,169,009,032,210,255,243
49230 :076,121,008,169,227,160,071
49236 :012,032,030,171,169,008,250
49242 :133,002,032,067,012,165,245
49248 :144,048,021,169,008,160,134
49254 :013,032,030,171,169,009,014
49260 :133,002,032,067,012,165,007
49266 :144,040,003,076,025,009,163
49272 :169,003,032,195,255,169,175
49278 :004,032,195,255,169,005,010
49284 :032,195,255,169,006,032,053
49290 :195,255,169,013,032,195,229
49296 :255,169,015,032,195,255,041
49302 :032,204,255,169,206,160,152
49308 :015,032,030,171,169,015,076
49314 :168,162,008,032,186,255,205
49320 :169,000,032,189,255,032,077
49326 :192,255,162,015,032,198,004
49332 :255,032,207,255,201,013,119
49338 :240,014,201,032,240,004,149
49344 :201,065,144,241,032,210,061
49350 :255,076,182,000,169,015,135
49356 :032,195,255,032,204,255,153
49362 :032,231,255,169,224,160,001
49368 :015,032,030,171,169,015,136
49374 :168,162,009,032,186,255,010
49380 :169,000,032,189,255,032,137
49386 :192,255,162,015,032,198,064
49392 :255,032,207,255,201,013,179
49398 :240,014,201,032,240,004,209
49404 :201,065,144,241,032,210,121
49410 :255,076,242,000,169,015,255
49416 :032,195,255,032,204,255,213
49422 :169,013,032,210,255,162,087
49428 :120,108,000,003,169,087,003
49434 :160,013,032,030,171,169,009
49440 :015,168,162,008,032,186,091
49446 :255,169,000,032,189,255,170
49452 :032,192,255,169,013,160,105
49458 :162,008,032,186,255,169,094
49464 :001,162,224,160,012,032,135
49470 :189,255,032,192,255,162,123
49476 :015,032,201,255,162,000,221
49482 :189,192,013,032,210,255,196
49488 :232,224,013,208,245,032,010
49494 :204,255,162,013,032,198,102
49500 :255,162,000,032,207,255,235
49506 :157,243,015,232,200,247,176
49512 :032,204,255,162,015,032,036
49518 :140,012,165,167,240,003,077
49524 :076,121,008,169,013,032,023
49530 :195,255,169,015,032,195,215
49536 :255,169,123,160,013,032,112
49542 :030,171,169,010,032,210,252
49548 :255,162,144,189,243,015,124
49554 :032,210,255,232,224,164,239
49560 :200,245,169,013,032,210,005
49566 :255,169,146,160,013,032,165
49572 :030,171,160,000,132,006,151
49578 :169,004,133,165,165,176,214
49584 :133,251,165,177,133,252,007
```

```
49590 :165,178,133,003,165,179,237
49596 :133,004,032,165,012,162,184
49602 :000,169,001,133,005,165,163
49608 :166,037,005,240,006,169,055
49614 :046,145,251,200,004,169,005
49620 :081,145,251,032,093,012,058
49626 :202,200,234,032,165,012,047
49632 :162,000,169,001,133,005,190
49638 :165,166,037,005,240,006,081
49644 :169,046,145,251,208,004,035
49650 :169,081,145,251,032,093,245
49656 :012,202,200,234,032,165,077
49662 :012,162,005,169,001,133,224
49668 :005,165,166,037,005,240,110
49674 :006,169,046,145,251,200,067
49680 :004,169,081,145,251,032,186
49686 :093,012,202,208,234,230,233
49692 :165,230,176,230,178,230,213
49698 :006,165,006,201,035,176,111
49704 :002,144,131,169,056,160,190
49710 :014,032,030,171,160,144,085
49716 :162,000,185,243,015,157,046
49722 :207,012,232,200,224,016,181
49728 :208,244,032,204,255,169,152
49734 :015,160,162,009,032,186,130
49740 :255,169,000,032,189,255,208
49746 :032,192,255,162,015,032,002
49752 :201,255,162,000,189,204,075
49758 :012,032,210,255,232,224,035
49764 :019,200,245,032,204,255,039
49770 :162,015,032,148,012,165,128
49776 :167,240,003,076,121,008,215
49782 :169,015,032,195,255,032,048
49788 :204,255,169,003,162,008,157
49794 :160,015,032,186,255,169,179
49800 :000,032,189,255,032,192,068
49806 :255,169,005,162,009,160,134
49812 :015,032,186,255,169,000,037
49818 :032,189,255,032,192,255,085
49824 :169,004,168,162,008,032,191
49830 :186,255,169,001,162,224,139
49836 :160,012,032,189,255,032,084

49842 :192,255,169,006,168,162,106
49848 :009,032,186,255,169,001,068
49854 :162,224,160,012,032,189,201
49860 :255,032,192,255,160,000,066
49866 :132,006,169,004,133,165,043
49872 :169,001,133,249,169,124,029
49878 :133,176,169,004,133,177,238
49884 :169,000,133,250,165,176,089
49890 :133,251,165,177,133,252,057
49896 :162,016,177,251,201,081,096
49902 :208,017,165,197,201,004,006
49908 :208,002,240,091,032,148,197
49914 :011,165,167,240,002,208,019
49920 :082,032,126,012,202,208,150
49926 :227,165,249,201,018,144,242
49932 :014,201,025,144,014,201,099
49938 :031,144,014,201,036,144,076
49944 :014,240,056,162,005,208,197
49950 :010,162,003,208,006,162,069
49956 :002,208,002,162,001,177,076
49962 :251,201,081,208,017,165,197
49968 :197,201,004,208,002,240,132
49974 :028,032,148,011,165,167,093
49980 :240,002,208,019,032,126,175
49986 :012,202,208,227,230,176,097
49992 :230,249,165,249,201,036,178
49998 :240,003,076,221,010,169,029
50004 :004,032,195,255,169,006,233
50010 :032,195,255,169,003,032,008
50016 :195,255,169,005,032,195,179
50022 :255,032,204,255,165,167,156
50028 :208,031,169,000,133,002,147
50034 :032,067,012,169,009,133,024
50040 :002,032,067,012,169,044,190
50046 :160,013,032,030,171,165,105
50052 :197,201,005,240,007,201,215
50058 :004,208,246,076,121,000,033
50064 :076,029,000,152,072,138,107
50070 :072,165,249,032,175,012,087
50076 :165,169,141,011,015,141,030
50082 :023,015,141,149,007,165,150
50088 :170,141,012,015,141,024,159
50094 :015,141,150,007,165,250,134
50100 :032,175,012,165,169,141,106
50106 :014,015,141,026,015,141,026
50112 :189,007,165,170,141,015,111
50118 :015,141,027,015,141,190,215

50124 :007,032,254,237,162,003,131
50130 :032,201,255,162,000,189,025
50136 :004,015,032,210,255,232,196
50142 :224,012,208,245,032,254,173
50148 :237,162,004,032,198,255,092
50154 :162,000,032,173,241,157,231
50160 :243,015,232,208,247,032,193
50166 :254,237,162,003,032,148,058
50172 :012,165,167,240,003,076,147
50178 :059,012,032,254,237,162,246
50184 :006,032,201,255,162,001,153
50190 :189,243,015,032,221,237,183
50196 :232,208,247,189,243,015,130
50202 :032,221,237,032,254,237,015
50208 :162,005,032,201,255,162,081
50214 :000,189,016,015,032,210,244
50220 :255,232,224,012,208,245,196
50226 :032,204,255,162,005,032,228
50232 :148,012,032,204,255,104,043
50238 :170,104,168,096,169,015,016
50244 :168,166,002,032,186,255,109
50250 :169,002,162,225,160,012,036
50256 :032,189,255,032,192,255,011
50262 :169,015,032,195,255,096,080
50268 :169,000,145,003,024,165,086
50274 :251,105,040,133,251,165,019
50280 :252,105,000,133,252,024,102

50286 :165,003,105,040,133,003,047
50292 :165,004,105,000,133,004,015
50298 :006,005,096,177,251,009,154
50304 :128,145,251,230,250,024,132
50310 :165,251,105,040,133,251,055
50316 :165,252,105,000,133,252,023
50322 :096,169,000,133,167,032,231
50328 :198,255,032,207,255,201,020
50334 :050,144,002,230,167,096,079
50340 :230,165,166,165,109,243,042
50346 :015,133,166,096,162,048,022
50352 :056,233,010,144,003,232,086
50358 :176,249,105,058,134,169,049
50364 :133,170,096,085,049,058,011
50370 :049,051,032,048,032,049,199
50376 :056,032,048,078,048,058,008
50382 :032,032,032,032,032,032,142
50388 :032,032,032,032,032,032,148
50394 :032,032,032,032,032,035,157
50400 :073,048,013,013,032,032,179
50406 :201,078,073,084,073,065,036
50412 :076,073,090,073,078,071,185
50418 :032,083,079,085,082,067,158
50424 :069,032,068,073,083,075,136
50430 :032,068,082,073,086,069,152
50436 :046,013,000,013,032,032,140
50442 :201,078,073,084,073,065,072
50448 :076,073,090,073,078,071,221
50454 :032,068,069,083,084,073,175
50460 :078,065,084,073,079,078,229
50466 :032,068,082,073,086,069,188
50472 :046,013,000,147,080,082,152
50478 :069,083,083,032,070,051,178
50484 :032,084,079,032,067,079,169
50490 :080,089,032,065,078,079,225
50496 :084,072,069,082,044,032,191
50502 :078,049,032,084,079,032,160
50508 :081,085,073,084,046,046,235
50514 :046,009,146,000,013,032,072
50520 :032,210,069,065,060,073,093

50526 :078,071,032,194,193,205,099
50532 :044,032,070,079,082,077,228
50538 :065,084,084,073,078,071,049
50544 :032,065,082,067,072,073,247
50550 :006,069,046,000,147,142,096
50556 :008,213,195,195,195,201,107
50562 :084,082,065,067,075,213,204
50568 :195,195,195,195,195,195,026
50574 :195,195,000,221,032,032,049
50580 :032,032,032,032,032,032,084
50586 :032,032,032,032,049,032,107
50592 :032,032,032,032,032,032,096
50598 :032,032,050,032,032,032,120
50604 :032,032,032,032,032,032,108
50610 :051,032,032,032,032,032,133
50616 :013,221,032,032,032,049,051
50622 :050,051,052,053,054,055,249
50628 :056,057,048,049,050,051,251
50634 :052,053,054,055,056,057,017
50640 :048,049,050,051,052,053,255
50646 :054,055,056,057,048,049,021
50652 :050,051,052,053,013,202,129

50658 :032,032,040,013,083,032,210
50664 :032,049,013,069,032,032,203
50670 :050,013,067,032,032,051,227
50676 :013,084,032,032,052,013,214
50682 :079,032,032,053,013,082,029
50688 :032,032,054,013,213,032,120
50694 :032,055,013,221,032,032,135
50700 :056,013,221,032,032,057,167
50706 :013,221,032,049,048,013,138
50712 :221,032,049,049,013,221,097
50718 :032,049,050,013,221,032,171
50724 :049,051,013,221,032,049,195
50730 :052,013,221,032,049,053,206
50736 :013,221,032,049,054,013,174
50742 :000,221,032,049,055,029,104
50748 :029,029,029,029,029,029,234
50754 :029,029,029,029,029,029,240
50760 :029,029,029,029,029,029,246
50766 :029,029,029,029,029,029,252
50772 :029,029,029,029,029,213,186
50778 :195,195,195,195,013,221,080
50784 :032,049,056,029,029,029,064
50790 :029,029,029,029,029,029,020
50796 :029,029,029,029,029,029,026
50802 :029,029,029,029,029,029,032
50808 :029,029,029,213,195,195,042
50814 :195,195,195,203,032,032,210
50820 :032,032,013,221,032,049,255
50826 :057,029,029,029,029,029,084
50832 :029,029,029,029,029,029,062
50838 :029,029,029,029,029,029,068
50844 :213,195,195,195,195,195,064
50850 :195,203,032,032,084,082,022
50856 :065,067,075,032,048,049,248
50862 :013,221,032,050,048,029,055
50868 :029,029,029,029,029,029,098
50874 :029,029,029,029,029,029,104
50880 :029,029,029,029,221,018,035
50886 :065,082,067,072,073,086,131
50892 :069,146,032,083,069,067,158
50898 :084,079,082,032,048,048,071
50904 :013,202,195,195,195,195,187
50910 :195,195,195,195,195,195,112
50916 :195,195,195,195,195,195,118
50922 :195,195,195,195,203,018,211
50928 :080,072,073,076,032,078,139
50934 :069,076,083,079,078,032,151
50940 :032,049,057,056,053,019,006
50946 :000,085,049,058,052,032,022
50952 :048,032,048,049,032,048,009
50958 :048,085,050,058,054,032,085
50964 :048,032,048,049,032,048,021
50970 :048,147,008,014,032,032,051
50976 :018,042,032,193,210,195,210
50982 :200,201,214,197,032,042,156
50988 :042,032,212,087,079,032,016
50994 :196,082,073,086,069,032,076
51000 :195,079,080,073,069,082,122
51006 :032,042,013,013,032,032,226
51012 :208,085,084,032,083,079,127
51018 :085,082,067,069,032,068,221
51024 :073,083,075,032,073,078,238
51030 :032,068,082,073,086,069,240
51036 :032,069,073,071,072,084,237
51042 :044,013,032,032,070,079,112
51048 :082,077,065,084,084,069,053
51054 :068,032,068,073,083,075,253
51060 :032,073,070,032,060,082,225
51066 :073,086,069,032,070,073,021
51072 :078,069,046,013,013,032,123
51078 :032,208,082,069,083,083,179
51084 :032,018,070,055,146,032,237
51090 :070,085,070,067,084,073,091
51096 :079,070,032,075,069,089,062
51102 :032,004,079,032,066,069,008
51108 :071,073,070,044,013,032,219
51114 :032,010,070,049,146,032,005
51120 :075,069,009,032,084,079,092
51126 :032,065,066,079,002,084,078
51132 :032,070,079,087,032,079,063
51138 :002,032,076,065,084,069,090
51144 :002,046,046,046,000,147,055
51150 :010,083,079,005,082,067,100
51156 :069,032,068,082,073,086,110
51162 :069,032,013,013,000,013,102
51168 :013,018,065,082,067,072,029
51174 :073,006,069,032,060,002,120
51180 :073,086,069,013,013,000,234
```

Atari Color Mixing

Karl E. Wiegers

This informative tutorial demonstrates the principles of color mixing on Atari 400/800, XL, and XE computers.

Advertisements for home computers often tout the number of different colors that a particular machine can produce. But little is said about how these colors are generated on your TV or monitor screen. Knowing some theory behind these techniques can help you produce more colorful graphics displays.

Atari computers manufactured after early 1982 can generate 256 color variations, based on 16 different hues which each can have 16 luminances (brightnesses). Ataris made before early 1982 can display only 8 luminances per hue for a total of 128 colors, but can be upgraded by replacing the CTIA chip with a GTIA chip. Atari colors are represented by a number from 0–255 using this formula: *color number = hue number * 16 + luminance number.*

The "Atari Color Mixing" program listed below demonstrates *additive* color mixing and lets you try your hand at *subtractive* color mixing as well. We'll explain these terms in detail in a moment. For now, enter and save Color Mixing, then run it. The program is quite simple and contains all the instructions you need. Just follow the prompts and refer to the rest of this article for additional information.

The colors may look better if you adjust your TV's brightness control somewhat higher than usual. The exact hues may also vary depending on the tint setting. Adjust the tint for good green and red, and the other colors should be pretty close.

Additive Color Mixing

As every child discovers when painting with watercolors, the three primary colors—red, green, and blue—can be added together in various combinations to make many different colors. Thus, red, green, and blue are known as the *additive primary colors.*

When primary colors are combined, new colors appear. White light is a balanced combination of red, green, and blue light. Equal intensities of blue and green light produce a greenish-blue color called cyan. Red and blue light mix to create magenta, a reddish-purple. And, believe it or not, mixing red and green light together produces yellow. Many more colors can be produced if the primaries are of different intensities. For instance, combining a given intensity of green with twice that intensity of red is equivalent to yellow plus red, or orange.

Additive color mixing works with pigments as well as lights. However, pigment mixing often results in different colors than those described above. For example, mixing red and green paints usually produces brown, not yellow.

In fact, the brown color really *is* a yellow. But red and green pigments usually have far less intensity than red and green lights. Besides having a particular *hue* (light frequency), a color can have different levels of *luminance* (intensity). The effect we usually call *color* is actually the combined effects of hue and luminance. Because red and green pigments are quite dark (have little luminance), mixing them together produces the dark yellow we commonly call brown.

A color TV or monitor creates different colors by the additive process using colored light sources. The screens of color TV tubes contain thousands of tiny red, green, and blue dots (phosphors) which glow when struck by electrons from a gun at the back of the tube. If you examine a color screen with a magnifying glass, you'll see individual red, green, and blue phosphors. At normal viewing distances the colored dots merge together and create additive colors. For instance, adjacent red and blue dots look like magenta. When no phosphors are lit, the result is black.

Subtractive Color Mixing

Recall that if red and blue lights are combined, the resulting color is magenta. There's also another way to produce magenta—you can shine white light through a magenta-colored filter. White light contains all the primary colors, but the filter absorbs the green light, allowing only the red and blue light to pass. In other words, a magenta filter *subtracts* or blocks out green light. If you place a green filter and a magenta filter in front of a white light source, all light should be blocked out: The green filter blocks red and blue, and the magenta filter blocks green. The final result is black. For this reason, green and magenta are termed *complementary colors* (magenta is also sometimes called *minus green*).

Similar logic applies to the other primary colors: A cyan (blue + green) filter subtracts red light, and a yellow (red + green) filter sub-

tracts blue light. Red and cyan (minus red) are complementary, as are blue and yellow (minus blue).

Cyan, magenta, and yellow are called *subtractive primary colors.* Just like the additive primaries (red, green, blue), the subtractive primaries can be mixed into virtually any combination of hue and luminance. But the process is reversed. Additive color mixing works by *sending* specific colors to your eye, while the subtractive process *removes* specific colors from a color-rich light source, leaving only complementary colors. Most color photographic systems are subtractive, using cyan, magenta, and yellow film dyes.

Color Mixing uses color numbers which—on my system—come closest to producing the six additive and subtractive primary colors (see lines 270–280). Of course, colors can vary greatly from one TV or monitor to the next. Cyan is a little difficult to display; my choice for cyan would look a bit greener, but my computer won't cooperate. The blue is also darker than you might expect, but blue in a color mixing sense is actually quite dark.

Additional Techniques

Note that each display screen in Color Mixing uses several different Atari graphics modes. The heading is displayed in graphics mode 1, other text is in mode 0, and the color squares are drawn in mode 3. Mixed-mode screens like this are created by modifying the computer's *display list,* a set of instructions which tells the computer how to put data on the screen. The Color Mixing program modifies display lists in lines 210, 1010–1020, 1210–1240, and 4010–4020. (You can read more about modifying display lists in *COMPUTE!'s First Book of Atari* and *COMPUTE!'s First Book of Atari Graphics.*)

Atari computers can ordinarily display up to five colors at a time. But some of the screens in Color Mixing show nine colors. This is accomplished with a *display list interrupt* (DLI). A DLI is a short machine language routine that, among other things, can change the contents of color registers while the computer is displaying each video frame. This technique lets you create graphics with extra colors on various parts of the screen. (For more information, consult *De Re Atari,* published by Atari, Inc.)

The program's colored boxes are drawn in graphics mode 3 using character strings for graphics storage (lines 310–320). Here is the text equivalent of this display:

```
AAAAA     CCCCC     BBBBB
AAAAA     CCCCC     BBBBB
AAAAA     CCCCC     BBBBB
AAAAA     CCCCC     BBBBB
```

When the computer displays a string in a nontext graphics mode with the PRINT#6 statement, the letters A, B, and C show up as different-colored pixels. The letter A appears as a pixel with the color taken from color register 0; the letter B uses color register 1; and the letter C uses register 2. A SETCOLOR or POKE statement which changes the value in color register 0, for example, would change the color of the A box. This technique is used in lines 4170 and 4210–4220. Line 4270 erases the boxes by setting all their colors to black. As Color Mixing demonstrates, it's far more convenient to store these graphics in strings than to use PLOT, DRAWTO, or XIO fill statements.

Atari Color Mixing

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
AJ 10  DIM A$(5),B$(5),C$(5),
       D1$(60),D2$(180),KDLOR
       S(6,6),BL$(35)
KH 15  DIM REO$(1),GREEN$(1),
       BLUE$(1),CYAN$(1),MAGE
       NTA$(1),YELLOW$(1),BLA
       CK$(1)
GJ 20  A$="AAAAA":B$="BBB88":
       C$="CCCCC"
IB 25  BL$(1)=CHR$(32):BL$(35
       )=BL$:BL$(2)=BL$
FB 30  GRAPHICS 0:? :? "DNE M
       OMENT..."
NB 40  FOR I=1 TO 30:READ A:D
       1$(I,I)=CHR$(A):NEXT I
       :D1$(31)=D1$
FM 50  DATA 72,138,72,169,66,
       162,180,141,10,212,141
       ,23,208
LI 60  DATA 142,24,208,169,0,
       141,0,2,169,0,141,1,2
DB 70  DATA 104,170,104,64
CB 80  D1$(35,35)=CHR$(10):D1
       $(37,37)=CHR$(0)
MK 90  D2$=D1$:D2$(61)=D1$:D2
       $(121)=D1$
IE 100 ST=ADR(D1$):GOSUB 400
DK 110 D1$(48,48)=CHR$(LO):D
       1$(53,53)=CHR$(HI)
FK 120 ST=ST+30:GOSUB 400
DA 130 D1$(18,18)=CHR$(LO):D
       1$(23,23)=CHR$(HI)
IJ 140 ST=ADR(D2$):GOSUB 400
LM 150 D2$(168,168)=CHR$(LO)
       :D2$(173,173)=CHR$(HI
       )
EL 160 FOR I=18 TO 138 STEP
       30
PP 170 ST=ST+30:GOSUB 400
CP 180 D2$(I,I)=CHR$(LO):D2$
       (I+5,I+5)=CHR$(HI)
CC 190 NEXT I
MI 200 D2$(132,132)=CHR$(22)
MF 210 DL=PEEK(560)+256*PEEK
       (561)
LB 220 RESTORE 240:FOR I=1 T
       O 6:FOR J=1 TO 6
BJ 230 READ A:KOLORS(I,J)=A:
       NEXT J:NEXT I
MC 240 DATA 66,26,86,4,82,40
       ,26,180,116,164,4,200
ME 250 DATA 86,116,146,150,1
       02,4,4,164,150,116,14
       6,180
ND 260 DATA 82,4,102,146,86,
       66,40,200,4,180,66,26
DK 270 REO$=CHR$(66):GREEN$=
       CHR$(180):BLUE$=CHR$(
       146)
BD 280 CYAN$=CHR$(116):MAGEN
       TA$=CHR$(86):YELLOW$=
       CHR$(26):BLACK$=CHR$(
       0)
JG 290 GOTO 2000
NF 300 POKE 87,3:GOSUB 500
NB 310 FOR I=0 TO 3:POSITION
       5,I:? #6;A$:POSITION
       16,I:? #6;C$
DC 320 POSITION 31,I:? #6;B$
       :NEXT I:RETURN
PL 350 POKE 87,0:GOSUB 500:R
       ETURN
GJ 400 HI=INT(ST/256):LO=ST-
       256*HI:RETURN
JM 500 HMEM=256*HMEM+LMEM+8Y
       TE
EI 510 LMEM=HMEM-256*INT(HME
       M/256)
CK 520 HMEM=INT(HMEM/256)
NF 530 POKE 88,LMEM:POKE 89,
       HMEM:RETURN
MG 600 OPEN #1,4,0,"K:":GET
       #1,A:CLOSE #1:RETURN
FD 1000 POKE 559,0:POKE 752,
       1
NP 1010 PDKE OL+3,70:PDKE DL
       +6,134:PDKE DL+14,13
       6
AB 1020 FOR I=7 TO 13:POKE O
       L+I,8:NEXT I
DO 1030 ST=AOR(D1$):GOSUB 40
       0:PDKE 512,LO:POKE 5
       13,HI
EJ 1040 LMEM=PEEK(88):HMEM=P
       EEK(89):PDKE 710,0:P
       DKE 54286,192:RETURN

HI 1200 POKE 559,0:POKE 752,
       1:RESTORE 1220
KD 1210 FOR I=6 TO 24:READ A
       :PDKE DL+I,A:NEXT I
BB 1220 OATA 134,8,8,8,136,2
       ,2,130,8,8,8
OI 1230 OATA 136,2,2,130,8,8
       ,8,136
HH 1240 POKE OL+3,70
ED 1250 ST=AOR(D2$):GDSUB 40
       0:PDKE 512,LO:PDKE 5
       13,HI
EN 1260 LMEM=PEEK(88):HMEM=P
       EEK(89):PDKE 710,0:P
       OKE 54286,192:RETURN

BP 2000 GRAPHICS 0:GOSUB 100
       0:PDKE 708,146:POKE
       711,40
```

```
KD 2010 O1$(5,5)=GREEN$:O1$(
        7,7)=RED$:POKE 559,3
        4
DJ 2020 POKE B7,1:BYTE=0:GOS
        UB 500
BA 2030 POSITION 6,0:? #6;"E
        [illegible]":? #6;"
        (3 SPACES)[illegible] CO
        LORS"
CB 2040 POKE B7,3:BYTE=40:GO
        SUB 500
MF 2050 FOR I=2 TO 5:POSITIO
        N 5,I:? #6;A$:POSITI
        ON 17,I:? #6;B$:POSI
        TION 30,I:? #6;C$:NE
        XT I
CE 2060 POKE B7,0:BYTE=B0:GO
        SUB 500
AA 2070 POSITION 5,0:? "BLUE
        (B SPACES)GREEN
        (9 SPACES)RED"
NB 20B0 POSITION 2,4
LD 2090 ? "These are the [illegible]
        [illegible] PRIMARY":? "co
        lors. The eye sees t
        he combination"
GL 2100 ? "of blue, green, a
        nd red as white.":?
        "Pairs of these prim
        aries are"
KI 2110 ? "perceived as new
        colors, the":? "[illegible]
        [illegible]TIVE PRIMARY colo
        rs."
BK 2120 POSITION 5,13:? "[illegible]
        URN TO GO ON, [illegible] TO
         QUIT";
LE 2130 GOSUB 600:IF A=155 T
        HEN 2500
PD 2140 IF A=27 THEN GRAPHIC
        S 0:END
MH 2150 GOTO 2130
CB 2500 GRAPHICS 0:GOSUB 120
        0:POKE 70B,146:POKE
        711,40
FH 2530 O2$(5,5)=MAGENTA$:O2
        $(7,7)=RED$:O2$(65,6
        5)=CYAN$
EL 2540 O2$(67,67)=GREEN$:O2
        $(125,125)=RED$
FB 2550 O2$(95,95)=YELLOW$:O
        2$(127,127)=GREEN$
DC 2560 POKE 559,34
PD 2570 POKE B7,1:BYTE=0:GOS
        UB 500
JK 25B0 POSITION 3,0:? #6;"[illegible]
        [illegible]"
EE 2590 BYTE=40:GOSUB 300:GO
        SUB 350
II 2600 POSITION 5,1:? #6;"B
        LUE(3 SPACES)+
        (4 SPACES)RED
        (5 SPACES)=
        (4 SPACES)MAGENTA"
HL 2610 BYTE=120:GOSUB 300:B
        YTE=40:GOSUB 350
EO 2620 POSITION 5,1:? #6;"B
        LUE(3 SPACES)+
        (3 SPACES)GREEN
        (4 SPACES)=
        (5 SPACES)CYAN"
HN 2630 BYTE=120:GOSUB 300:B
        YTE=40:GOSUB 350
LF 2640 POSITION 6,1:? #6;"R
        ED(3 SPACES)+
        (3 SPACES)GREEN
        (4 SPACES)=
        (5 SPACES)YELLOW"
IB 2660 POSITION 5,3:? "[illegible]
        RN TO GO ON, [illegible] TO
        REVIEW";
LJ 2670 GOSUB 600:IF A=155 T
        HEN 3000
DH 26B0 IF A=27 THEN 2000
NJ 2690 GOTO 2670
DN 3000 GRAPHICS 0:GOSUB 100
        0:POKE 70B,26:POKE 7
        11,40
IL 3010 O1$(5,5)=MAGENTA$:O1
        $(7,7)=CYAN$:POKE 55
        9,34
DK 3020 POKE B7,1:BYTE=0:GOS
        UB 500
PC 3030 POSITION 5,0:? #6;"[illegible]
        [illegible]TIVE":? #6;"
        (3 SPACES)[illegible] CO
        LORS"
CC 3040 POKE B7,3:BYTE=40:GO
        SUB 500
MG 3050 FOR I=2 TO 5:POSITIO
        N 5,I:? #6;A$:POSITI
        ON 17,I:? #6;B$:POSI
        TION 30,I:? #6;C$:NE
        XT I
CF 3060 POKE B7,0:BYTE=B0:GO
        SUB 500
JA 3070 POSITION 4,0:? "YELL
        OW(6 SPACES)MAGENTA
        (B SPACES)CYAN"
NC 30B0 POSITION 2,4
IM 3090 ? "These are the [illegible]
        [illegible]TIVE PRIMARY":?
        "colors. A yellow fi
        lter subtracts"
LI 3100 ? "the blue componen
        t out of white":? "l
        ight, only passing t
        he red and"
EC 3110 ? "green components.
          Yellow and blue":?
        "are thus called [illegible]
        [illegible]TIVE COLORS."
KK 3120 POSITION 5,13:? "[illegible]
        URN TO GO ON, [illegible] TO
        REVIEW";
LG 3130 GOSUB 600:IF A=155 T
        HEN 3500
DE 3140 IF A=27 THEN 2500
NJ 3150 GOTO 3130
CE 3500 GRAPHICS 0:GOSUB 120
        0:POKE 70B,116:POKE
        711,40
DL 3530 O2$(5,5)=BLUE$:O2$(7
        ,7)=MAGENTA$:O2$(65,
        65)=GREEN$
NJ 3540 O2$(67,67)=YELLOW$:O
        2$(125,125)=MAGENTA$
LO 3550 O2$(95,95)=YELLOW$:O
        2$(127,127)=RED$
OD 3560 POKE 559,34
PE 3570 POKE B7,1:BYTE=0:GOS
        UB 500
HL 35B0 POSITION 1,0:? #6;"[illegible]
        [illegible]TIVE COLORS"
EF 3590 BYTE=40:GOSUB 300:GO
        SUB 350
NJ 3600 POSITION 5,1:? #6;"C
        YAN(3 SPACES)+  MAGE
        NTA(4 SPACES)=
        (4 SPACES)BLUE"
HM 3610 BYTE=120:GOSUB 300:B
        YTE=40:GOSUB 350
AD 3620 POSITION 5,1:? #6;"C
        YAN(3 SPACES)+
        (3 SPACES)YELLOW
        (4 SPACES)=
        (4 SPACES)GREEN"
AD 3630 POKE B7,3:BYTE=120:G
        OSUB 500:FOR I=0 TO
        3
BN 3634 POSITION 5,I:? #6;A$
        :POSITION 16,I:? #6;
        B$:POSITION 31,I:? #
        6;C$:NEXT I
PL 363B BYTE=40:GOSUB 350
EA 3640 POSITION 4,1:? #6;"M
        AGENTA  + YELLOW
        (4 SPACES)=
        (5 SPACES)RED"
IC 3660 POSITION 5,3:? "[illegible]
        RN TO GO ON, [illegible] TO
        REVIEW";
LL 3670 GOSUB 600:IF A=155 T
        HEN 4000
DJ 36B0 IF A=27 THEN 3000
NL 3690 GOTO 3670
MA 4000 GRAPHICS 0:GOSUB 100
        0
PF 4010 FOR I=6 TO B:POKE OL
        +I,6:NEXT I:POKE OL+
        9,134
HP 4020 FOR I=10 TO 14:POKE
        OL+I,B:NEXT I:POKE O
        L+15,136
ID 4030 POKE 711,40:POKE 70B
        ,0
IC 4040 O1$(5,5)=BLACK$:O1$(
        7,7)=BLACK$
CD 4050 POKE 559,34
OP 4060 POKE B7,1:BYTE=0:GOS
        UB 500
PD 4070 POSITION 2,1:? #6;"[illegible]
        [illegible]TIVE COLOR":PO
        SITION 3,2:? #6;"[illegible]
        [illegible]"
HD 40B0 BYTE=100:GOSUB 300:B
        YTE=60:GOSUB 350
HM 4090 POSITION 12,0:? #6;"
        +":POSITION 24,0:? #
        6;"="
DJ 4100 POSITION 9,4:? #6;"1
        -RED(7 SPACES)4-CYAN
        "
FF 4110 POSITION 9,5:? #6;"2
        -GREEN(5 SPACES)5-MA
        GENTA"
DP 4120 POSITION 9,6:? #6;"3
        -BLUE(6 SPACES)6-YEL
        LOW"
KC 4140 POSITION 2,9:? #6;"E
        NTER A COLOR NUMBER:
         ";
KJ 4145 GOSUB 600:C1=A-4B
JJ 4150 IF C1<1 OR C1>6 THEN
         ? CHR$(253);:GOTO 4
        145
EJ 4170 ? #6;C1:POKE 70B,KOL
        ORS(C1,C1):POSITION
        7,0:? #6;C1
NG 41B0 POSITION 2,10:? #6;"
        MIX IT WITH COLOR NU
        MBER: ";
KD 41B5 GOSUB 600:C2=A-4B
KO 4190 IF C2<1 OR C2>6 THEN
         ? CHR$(253);:GOTO 4
        1B5
JD 4210 ? #6;C2:O1$(7,7)=CHR
        $(KOLORS(C2,C2)):POS
        ITION 1B,0:? #6;C2
BH 4220 O1$(5,5)=CHR$(KOLORS
        (C1,C2))
AK 4230 POSITION 6,12:? #6;"[illegible]
        [illegible] TO DO MORE, [illegible]
        [illegible] TO QUIT"
LO 4240 GOSUB 600:IF A=155 T
        HEN 4270
AC 4250 IF A=27 THEN GRAPHIC
        S 0:END
MP 4260 GOTO 4240
OL 4270 POKE 70B,0:O1$(5,5)=
        BLACK$:O1$(7,7)=BLAC
        K$
OM 42B0 POSITION 7,0:? #6;"
        ":POSITION 1B,0:? #6
        ;" "
IG 4300 FOR I=9 TO 12:POSITI
        ON 2,I:? #6;BL$:NEXT
         I:GOTO 4140
```

Mousor:
Escape Mode Cursor For The Apple IIc

J. Blake Lombert, Assistant Editor
Tim Victor, Editorial Programmer

This short, fast utility makes it simple to use your Apple IIc mouse controller for editing in BASIC or the machine language monitor in escape mode.

Despite all the improvements Apple incorporated into the IIc, the screen editing features when using BASIC or the machine language monitor are not much better than those available on the IIe. Without an editing support package, it is difficult to copy and correct program lines. And there is no way to use the mouse controller to make editing easier.

In BASIC, usually you end up making corrections by just typing the incorrect line all over again. This wastes time and effort. The alternative is to use what is called *escape mode* editing.

"Mousor" makes using escape mode easy. By rolling the mouse over an area of the desk smaller than a 3 × 5-inch index card, you can cursor (mousor) anywhere on the screen.

How To Use Mousor

To start mousing around with Mousor, type in and save the program below. It's a BASIC loader which creates the Mousor machine language routine in memory. (Note: Save the BASIC loader on disk before running it for the first time.) When you run Mousor, it automatically checks to see if you're using DOS 3.3 or ProDOS and then adjusts itself accordingly. When the BASIC prompt reappears, you'll have a mouse-driven escape mode cursor. If you don't understand escape mode editing, see the instructions below.

The mouse is trained to work like this:

1. Click the mouse button to activate escape mode.
2. While holding the button down, roll the mouse across the desk to move the escape mode cursor.
3. Release the mouse button to exit escape mode.

Mousor locks out keypresses while it is in escape mode, so if you want to use the escape editing functions (like ESC-E to erase the end of a line), press the ESC key.

Getting A Line Ot BASIC

When you type a line of BASIC on the Apple IIc, a routine called GETLN puts the characters into a special area of memory called the input buffer. The first character on the line is stored at the start of the input buffer, and subsequent characters are added to the end of the buffer. This continues until you press the RETURN key to enter the line (with a few important exceptions). The computer clears the rest of the current screen line and stores the carriage return character into the input buffer to mark the end of the line.

When you make a mistake while entering the line, like typing the wrong character, it's easy to fix. For example, if you are entering a line such as 10 PRINT "HELLP" and notice you pressed P instead of O, you can press the left-arrow key (also called backspace) to back up and change the letter. Instead of storing the backspace in the buffer like other keypresses, GETLN treats it differently.

GETLN keeps track of the length of the input buffer by pointing to the end. When GETLN receives a backspace character, it lowers the value of the pointer by one. This removes the last character in the buffer, so all you need to do is continue typing.

If you don't notice your mistake until you've typed in several more characters, you can use the left- and right-arrow keys to make the correction. Press the backspace key until the cursor is on the letter you want to change. Type the correct letter, and then press the right-arrow key (also called retype) until the cursor returns to the end of the line.

Each time you press the retype key, the character currently under the cursor is added to the input buffer and the cursor is moved to the right. In effect, you have removed several characters from the buffer, changed the character you wanted to correct, and retrieved the rest of the characters one by one from the screen.

Now For The Tricky Stuff

Unfortunately, you can't always catch your typing errors before you press RETURN. Often, you don't even know there's a problem in a line until you've run the program. Since the retype key allows you to pick up characters from the screen and add them to the input buffer, it would be handy if you could copy most of the bad line and type only the characters you want to change. This requires a way to move the cursor around the screen without affecting the input buffer.

Pressing the ESC key puts the IIc into escape mode. In this mode, the arrow keys move the cursor but don't change the input buffer. The IIc indicates escape mode by displaying a different cursor—an inverse plus sign. To leave escape mode, press ESC again.

Suppose you want to edit the following line in escape mode:

```
100 PRINT "THIS IS A TEDT"
```

If the line is not on the screen, you'll need to LIST 100. Press ESC to enter escape mode and move the cursor up to the 1. At this point, the input buffer is empty. Press ESC again and use the retype key to place the cursor on the D. This enters all but the last three characters into the input buffer. Now type the

letter S, and press retype twice, followed by RETURN. If you like, LIST 100 to verify the correction.

To edit the same line with Mousor, you would LIST the line, click the mouse button and drag the cursor to the 1, and release the mouse button. After this, follow the same editing procedure.

Mouse Moves

You can also use escape mode to grab pieces of program lines. Mousor is especially adept at this, since movement is so easy and quick. Just keep in mind that when the mouse button is down no characters are added to the buffer.

To copy a line, first LIST it. Then enter the number for the new line you want to create, click the mouse button and drag the escape mode cursor to just beyond the original line number, and release the button. Copy the line by pressing the right-arrow key until you reach the last character, and then press RETURN.

Inserting is another useful technique. LIST the line first, then mousor (click and drag the escape mode cursor) to the beginning of the line number. Release the mouse button and right-arrow across the line until you reach the point where you want to insert characters. Press the mouse button and mousor to a blank line on the screen, then release the button and type the insert characters. Click and drag up to the listed line again, release the button, and right-arrow to the end. After making any changes, don't forget to press the RETURN key to enter them.

Mousor For Apple IIc

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
DE 10 IF PEEK (191 * 256) = 76 T
      HEN GOSUB 40: GOSUB 50: GO
      TO 30
AC 20 GOSUB 50: GOSUB 40
2E 30 FOR I = 11 TO 207: READ A:
       POKE I + 768,A: NEXT : CA
      LL 768: END
69 40 FOR I = 0 TO 10: READ A: P
      OKE I + 768,A: NEXT : RETU
      RN
13 50 FOR I = 0 TO 10: READ A: N
      EXT : RETURN
E7 60 DATA 216,169,67,141,50,190
4E 70 DATA 169,3,141,51,190
BE 80 DATA 169,67,133,56,169,3
78 90 DATA 133,57,32,234,3
F7 100 DATA 120,162,196,160,64,3
      2
77 110 DATA 28,196,169,0,141,120
7C 120 DATA 4,141,120,5,141,248
EA 130 DATA 5,169,8,141,248,4
8F 140 DATA 169,0,162,196,160,64
4C 150 DATA 32,176,196,169,1,162
82 160 DATA 196,32,176,196,162,1
      96
EF 170 DATA 32,132,196,169,1,162
52 180 DATA 196,160,64,32,61,196
D6 190 DATA 88,96,145,40,32,76
66 200 DATA 204,44,99,192,16,8
51 210 DATA 32,112,204,16,246,76
2B 220 DATA 37,253,218,90,72,169
74 230 DATA 4,141,124,4,141,252
FC 240 DATA 4,32,187,3,32,195
FB 250 DATA 3,44,99,192,16,9
6A 260 DATA 104,32,179,195,122,2
      50
B2 270 DATA 76,69,3,173,124,4
B4 280 DATA 240,7,201,8,144,25
5C 290 DATA 162,156,44,162,136,1
      04
E5 300 DATA 32,179,195,138,32,88
3A 310 DATA 205,32,195,3,72,169
A3 320 DATA 4,141,124,4,32,187
49 330 DATA 3,173,252,4,240,7
DA 340 DATA 201,8,144,25,162,138
5F 350 DATA 44,162,159,104,32,17
      9
B2 360 DATA 195,138,32,88,205,32
0B 370 DATA 195,3,72,169,4,141
C7 380 DATA 252,4,32,187,3,76
1F 390 DATA 102,3,162,196,160,64
C9 400 DATA 32,107,196,96,32,29
60 410 DATA 204,72,41,128,73,171
15 420 DATA 32,179,195,104,96
```

Commodore 64 Headliner

Robert F. Lambiase

Create attention-getting headlines and titles with this oversized alphabet for the Commodore 64.

Nearly every program uses titles or headlines of some sort, and you ordinarily want titles to look as impressive as possible. But the standard Commodore character set doesn't permit much variety. You can use different character colors or print in reverse video, but the letters are still pretty small. "Commodore 64 Headliner" lets you create truly striking titles and headlines with an alphabet that's four times bigger than normal.

Enter and save Headliner from the listing below, then run it. After a short pause to form the new characters, the program prints the alphabet in giant, quadruple-size characters. The letters can be any color, and the standard-size alphabet is available, too. The only thing you give up are reverse video characters, since Headliner redefines them as large characters.

Using Headliner

Headliner is easy to incorporate in your own programs. The first step is to include lines 100–350 (they can be renumbered, of course) to create the new character set. The program begins storing the new character definitions at memory location 12288. The statement POKE 53272,29 (see line 240) tells the computer to look at this memory area for character definition data. Use POKE 53272,21 to switch back to normal characters.

After defining the new characters, Headliner prints the expanded alphabet (lines 400–405) and a title (lines 410–430). Whenever a string of large characters is to be printed, the characters are defined as a string (X$). Then two important variables (SL and CC) are defined. Finally, the statement GOSUB 500 calls the subroutine that puts the big characters on the screen. The subroutine analyzes each character in X$; if it is not a space character, its pattern is POKEd into screen memory.

The variable SL sets the position of the large characters on your screen. The 64's screen is divided into 25 rows and 40 columns, giving a total of 1,000 different loca-

tions. Each screen memory location has a different address, and they are numbered in order, beginning at the upper-left corner of the screen. The upper-left screen position is location 1024; the next location to the right is 1025, and so on. Color memory is a second 1,000-byte memory area that corresponds to screen memory. By POKEing the right number into color memory, you can control the color of any screen memory location. Your *Commodore 64 User's Guide* has maps that show the numbers for every screen memory and color memory location, as well as a list of all the color numbers.

To place large characters on the screen, find the location you want using the screen memory map in your user's guide, then set SL to that value. The upper-left corner of the first large character appears at the location defined by SL, and the others follow in order. For example, to start printing large characters at the upper-left corner of the screen, use the statement SL=1024.

Note that the title above the alphabet uses a large character to begin a line of standard characters, somewhat like a super-capital letter. Line 420 of the program sets the computer to start printing again at the next location *after* the last expanded character.

Pick Your Colors

The variable CC sets the color of the large characters using the color numbers listed in your user's guide. Line 410 of Headliner uses the statement CC=3 to print in cyan. Use the statement CC=1 to print in white, and so on.

There may be times when you want to print large characters in the current character color. Line 400 of Headliner does this with the statement CC=PEEK(646). Location 646 always contains the current color number for PRINTing characters.

The 64 actually has two alternate character sets: One is used in uppercase/graphics mode, and the other is used in lowercase/uppercase mode. You can switch from one mode to another by pressing SHIFT-COMMODORE. Since Headliner works only in uppercase/graphics mode, you should disable the SHIFT-COMMODORE key combination to prevent the user from accidentally destroying the display. To do this, insert PRINT CHR$(8) at the beginning of your program. When the program ends, type PRINT CHR$(9) to restore things to normal.

Building Giant Characters

Each large character is actually four redefined characters placed together. (To see this more clearly, type POKE 53272,21 and press RETURN after the program has run.) The standard Commodore 64 character set contains 256 characters, numbered from 0–255. Characters 0–127 are the "normal" characters and characters 128–255 are the same characters in reverse video. Since each character definition takes eight bytes, a full set of character definitions requires 2,048 (8*256) bytes.

The first step in redefining characters is to copy the standard character set from ROM (Read Only Memory) into RAM (Random Access Memory) where it can be altered. The program does this in lines 200–240. The new character set begins at location 12288. Since we only want characters 0–127 from the standard set, only those character definitions are copied.

Next the program POKEs the expanded character definitions into the memory area that would otherwise store reverse video character data (see lines 300–350). The bit pattern of each standard character is mapped into a four-character-sized memory area, using conversion values stored in the T() array.

Since each large character definition takes four times the memory of a standard definition, we have room for a maximum of 32 (128/4) expanded definitions. That's enough for 26 letters, but not enough space to hold ten numeral definitions, too. However, you could squeeze in six more characters—perhaps punctuation or other symbols.

By sacrificing all the standard characters, you can get as many as 64 large characters—but remember to define a space character so you can still clear the screen. The *Commodore 64 Programmer's Reference Guide* contains much more information about using redefined characters.

Headliner

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
100 DIMT(15):POKE53281,0:POKE5
    3280,0:PRINTCHR$(8):G=5427
    2                    :rem 24
110 POKE646,1:FORJ=0TO15:READT
    (J):NEXT             :rem 93
120 DATA 0,3,12,15,48,51,60,63
    ,192,195,204,207,240,243,2
    52,255               :rem 220
200 PRINTCHR$(147)TAB(125)"DOW
    NLOADING THE CHARACTER SET
    ":G=53248:GN=12288:rem 118
220 POKE 56333,127:POKE1,51:FO
    R Q=0TO1023:POKEGN+Q,PEEK(
    G+Q):NEXT            :rem 89
240 POKE1,55:POKE56333,129:POK
    E53272,29            :rem 196
300 PRINTCHR$(147)TAB(125)"FOR
    MING THE LARGE CHARACTERS"
    :POKE13312,0         :rem 15
320 FORR=0TO212STEP8:B1=12288+
    R:B2=13312+4*R       :rem 19
330 FORI=0TO4STEP4:FORK=0TO3:J
    =PEEK(B1+K+I):N=B2+2*(K+I)
    +1                   :rem 233
340 X1=T((JAND240)/16):X2=T(JA
    ND15)                :rem 197
350 POKEN,X1:POKEN+1,X1:POKEN+
    16,X2:POKEN+17,X2:NEXTK,I,
    R                    :rem 72
360 PRINTCHR$(147)       :rem 20
400 X$="ABCDEFGHIJKLM":SL=1270
    :CC=PEEK(646):GOSUB 500
                         :rem 110
405 X$="NOPQRSTUVWXYZ":SL=1350
    :CC=PEEK(646):GOSUB500
                         :rem 27
410 X$="H":SL=1158:CC=3:GOSUB
    {SPACE}500           :rem 62
420 NS=SL+40:POKE210,INT(NS/25
    6):POKE209,NS ANO 255
                         :rem 179
430 POKE646,7:PRINT"EADLINER":
    FORJ=1TO10:PRINT:NEXT:ENO
                         :rem 70
500 FORP=1TOLEN(X$):L=(ASC(MIO
    $(X$,P,1))-64)*4+128:IFL=0
    THEN550              :rem 111
530 POKESL+G,CC:POKESL+1+G,CC:
    POKESL+40+G,CC:POKESL+41+G
    ,CC                  :rem 137
540 POKESL,L:POKESL+1,L+2:POKE
    SL+40,L+1:POKESL+41,L+3
                         :rem 241
550 SL=SL+2:NEXT:RETURN :rem 5
```

Using The Commodore USR Function

Keith R. Bergerstock

The USR function provides a convenient way for BASIC programs to call machine language subroutines—and it's more versatile than the SYS statement. Although this article is oriented toward the Commodore 64, the general principles apply to all Commodore computers. A demonstration program shows how to add five new functions to Commodore 64 BASIC.

It's often overlooked, but the USR function is a powerful and convenient tool for accessing machine language (ML) routines from BASIC. In its simplest form, USR works just like the more familiar SYS command. SYS makes the computer halt BASIC program execution and jump to an ML routine at a specified address. When the ML routine is done, BASIC resumes what it was doing. SYS lets you jump anywhere in the computer's memory, to a system routine stored in Read Only Memory (ROM) or a user-written ML program stored in free memory.

To see an example on the Commodore 64, move the cursor to a blank line somewhere near the middle of the screen, type SYS 59626, and press RETURN. SYS 59626 jumps to the computer's ROM scrolling routine: The screen scrolls up and the blinking cursor reappears.

Although USR requires a little preparation, it's much easier to use after the preparation is done. Let's call the same scrolling routine with USR. Type the following line in direct mode (without a line number) and press RETURN.

```
POKE 785,234: POKE 786,232:
A=USR(0)
```

The screen scrolls upward, just as it did when you typed SYS 59626. The POKEs set up the routine's address for USR. This method looks cumbersome, but the POKEs are needed only once. Afterward you can call the scrolling routine whenever you like, just by entering A=USR(0). Program 1 below contains a formula that automatically performs the correct POKEs to prepare any address for USR.

Like PEEK and other BASIC functions, USR must be followed by a value in parentheses. However, in the simplest case (when you just want to jump to an ML routine), the value and the preceding variable name are both irrelevant: You get the same result with A=USR(XYZ) or GG=USR(123456 78). You can even use PRINT USR(X), though that usually prints something on the screen.

Parameter Passing

The real value of USR lies in its ability to pass parameters (values) back and forth between BASIC and machine language. To see how this works, type in and save Programs 1 and 2 below. Then run Program 1; it puts a short, multipurpose ML program in memory and sets up the USR *address vector* (a pair of memory locations that point to the ML routine).

The variable SA in line 10 defines the starting address of the ML routine. This ML program is relocatable, so you can put it elsewhere if you like. For instance, to put the routine at 49152, change line 10 so that SA=49152 and rerun Program 1.

The rest of line 10 converts the address into *low byte/high byte* format for the USR vector. Since 255 is the largest number any single memory location can hold, the computer must use two adjacent locations to store addresses like 59626. Program 1 stores the high byte of the address in the variable HI% and the low byte in LO.

Line 20 POKEs LO and HI% into vector locations 785 and 786. You must always put the target address in these locations before using USR. The rest of Program 1 POKEs the ML into the computer's memory. To use this technique in your own programs, just duplicate the method shown in Program 1.

If you get an ?ILLEGAL QUANTITY error message when experimenting with USR, it probably means that you forgot to put a vector address in 785–786. When you turn the computer on, the vector in 785–786 points to 45640, the BASIC routine that prints that message.

USR works virtually the same on all Commodore computers; the only difference is the location of the USR vector. You'll find it at locations 1–2 on the VIC-20 and 812–813 on the Commodore Plus/4 and 16. The other vectors mentioned below also are located in different places on various machines.

Using USR

Program 1 provides five handy functions which you select by inserting a number from 0–4 in the parentheses after USR. For an illustration, plug a joystick in port 2, then load and run Program 2 after running Program 1 to install the ML. As you move the joystick, the program prints the joystick directions on the screen. To exit the program, press the fire button.

Line 50 of Program 2 does the important work. The statement

JV=USR(3) calls the ML routine and selects function 3 (read joystick). Each time the ML routine performs this function, it gives the variable JV a numeric value representing the joystick position. JV is 0 when the joystick is centered, 9 when the fire button is pressed, and so on.

Note that Program 2 passes information in both directions. The value in parentheses—USR(3)—sends information to the ML routine (telling it to perform function 3). And the ML routine passes other information back *in the form of a variable* (JV).

The other four functions work in similar fashion. Function 4 returns the size of the BASIC program currently in memory. Whenever you want to know your program size, use the statements A=USR(4): PRINT A. Function 2 changes the screen background and border colors. To activate this function, use A=USR(C*256+2), replacing C with the number of the screen color you want (your *64 User's Guide* lists the color numbers).

Function 1 reads the Y and N keys, returning the value of 1 when Y is pressed, and 2 when N is pressed. The ML routine waits until you press Y or N, ignoring all other keys. This function is useful in the common case where a program asks the user a Yes/No question. Combining USR with ON-GOTO or ON-GOSUB is a very efficient technique. For instance, type in and run the following program (make sure the ML routine is in memory):

```
10 PRINT"ENTER Y/N":ON USR(1)
   GOSUB 100,200
20 GOTO 10
100 PRINT"YES":RETURN
200 PRINT"NO":RETURN
```

Function 0 is similar to Function 1, but reads the eight special function keys. Enter and run this program to see how it works:

```
10 A=USR(0)
20 PRINT A: GOTO 10
```

In this case USR returns a number from 1–8 in the variable A. (But note that the numbers returned don't correspond directly to the function key numbers. Keys f1, f3, f5, and f7 return values 1–4 respectively, while the shifted keys, f2, f4, f6, and f8, return values 5–8.) In your own programs, of course, you can use any variable name you like; this function could also be used with ON-GOTO or ON-GOSUB to select as many as eight different options.

The Facts About FAC1

If you simply want to use the new functions provided by Program 1, you needn't read any further. If you're ready to write your own ML routines for USR, here are a few additional tips. First, when you pass a value from BASIC, the value is converted into a different number format (floating point) and placed in the computer's *floating point accumulator* (locations 98–101).

The floating point accumulator—usually called FAC1 to distinguish it from the secondary accumulator—is a special number-processing area used internally for many purposes. Since floating point numbers are quite difficult to handle, it's helpful to convert the floating point value into an integer (whole number) before using it in your ML routine. Fortunately, the 64 has built-in routines to convert floating point numbers to integers and vice versa. These routines can be accessed directly with JSR, or indirectly through the vectors in locations 3–4 and 5–6.

The routine at location 45482 ($B1AA) converts a floating point number in FAC1 to an integer. Use this routine when passing a value from BASIC to ML. At the point where you want to retrieve the passed value, use JSR $B1AA to do the conversion. The computer returns the low byte of the integer in the Y register and the high byte in the A register. If you'd rather use the vector, use LDA #$4C: STA $02: JSR $0002.

Passing a value from ML back to BASIC often requires the opposite conversion. The routine at 45969 ($B391) converts an integer to floating point format and stores the result in FAC1. At the point where you want to return to BASIC, load the low byte of your integer value in the Y register and the high byte in A. Then call the integer-to-floating point conversion routine with JSR $B391: RTS (you can also compress these two instructions into JMP $B391). The value is converted and stored in FAC1, and RTS returns you to BASIC. If you prefer to use the vector, JMP ($0005) accomplishes the same thing.

Like other vectors, the vectors at 3–4 and 5–6 will presumably be safe to use even if the actual ROM addresses of the routines change after a ROM update. However, there's one danger in using them. Since BASIC never uses locations 2–6, many ML programmers use them as free zero page space. If your routine jumps through these vectors after some other ML program overwrites them, it may send the computer into never-never land.

Program 1: USR Loader For Commodore 64

```
10 SA=53088:HI%=SA/256:LO=SA-H
   I%*256                    :rem 23
20 POKE785,LO:POKE786,HI%:CK=0
                             :rem 33
30 READQ:IFQ>-1THENPOKESA,Q:SA
   =SA+1:CK=CK+Q:GOTO30
                             :rem 213
40 IFCK=14485THENPRINT"OK":NEW
                             :rem 130
50 PRINT"ERROR IN DATA STATEME
   NTS"                      :rem 121
60 DATA 169,0,133,198,169,76,1
   33,2,32,2,0,132,2         :rem 254
61 DATA 192,0,240,19,192,1,240
   ,35,192,2,240,50          :rem 197
62 DATA 192,3,240,71,192,4,240
   ,54,108,0,3,32,228        :rem 43
63 DATA 255,201,133,144,249,20
   1,141,176,245,56          :rem 218
64 DATA 233,132,168,169,0,108,
   5,0,32,228,255,201        :rem 51
65 DATA 89,240,8,201,78,208,24
   5,160,2,208,236           :rem 177
66 DATA 160,1,208,232,141,32,2
   08,141,33,208             :rem 54
67 DATA 169,0,168,108,5,0,56,1
   65,45,229,43,168          :rem 235
68 DATA 165,46,229,44,108,5,0,
   173,0,220,73,31           :rem 162
69 DATA 41,31,168,201,3,144,12
   ,136,201,8,144,7          :rem 204
70 DATA 136,201,16,144,2,160,9
   ,169,0,108,5,0,-1         :rem 236
```

Program 2: Joystick Demo For Commodore 64

```
10 DATA NONE,UP,DOWN,LEFT,UP/L
   EFT                       :rem 98
20 DATA DOWN/LEFT,RIGHT,UP/RIG
   HT                        :rem 54
30 DATA DOWN/RIGHT,FIRE BUTTON
                             :rem 144
40 PRINTCHR$(147):FORJ=0TO9:RE
   ADA$(J):SP$=SP$+CHR$(32):NE
   XT                        :rem 73
50 JV=USR(3):PRINTCHR$(19)JV,A
   $(JV)SP$                  :rem 44
60 IFJV=9THENE=E+1           :rem 93
70 IFE<20THEN50              :rem 109
```

Sound And Music On The Commodore 128 Part 1

Philip I. Nelson, Assistant Editor

The Commodore 128's advanced BASIC makes it easy and fun to create music or sound effects. Part 1 of this two-part series shows how to use the VOL, TEMPO, and ENVELOPE statements. Part 2 explores the FILTER, SOUND, and PLAY commands and includes three short tutorial programs.

If you've heard much about the new Commodore 128, you probably know that it contains a very powerful music maker: the SID (Sound Interface Device) chip, exactly as found in the Commodore 64 and still the best sound chip in any personal computer. The SID chip provides three independent voices (tone generators) for playing up to three notes at once, and four different waveforms to simulate virtually any sound.

Although both computers use the SID chip, the comparison ends there. Since Commodore 64 BASIC has no sound commands, even simple 64 sound effects require several POKE statements. The 128's BASIC eliminates the POKEs by adding six new music and sound commands: PLAY, SOUND, VOL, TEMPO, ENVELOPE, and FILTER.

Simplicity And Power

The PLAY command is both powerful and easy to use. If you have access to a 128, type in and run the following one-line program. (The spaces make the statement more readable, but are not necessary.)

```
100 PLAY "C D E F G F E D C"
```

The 128 plays nine notes, going up the scale and down again. It would take a lot more work to play the same nine notes on the 64—you'd need at least three preliminary POKEs (to set the volume and sound envelope), plus four POKEs *for each note* (one to turn on the voice, two to set the pitch, and one to turn off the voice).

Interestingly, you can control the SID chip in 128 mode with the same POKEs as on the 64. That's usually a waste of time, since the 128's BASIC commands are more convenient than POKEs. However, 128 BASIC has certain limitations (SOUND statements can't use ring modulation or synchronization, for example). If you already know sound programming on the 64, you may still find uses for old-fashioned 64 programming techniques.

The PLAY command is so versatile that it's almost a mini-language in itself. In addition to playing notes, you can insert rests, change octaves, choose any of ten different instrument voices, use filtering, and even play multivoice music. This month we'll stick to simple PLAY statements and examine the VOL, TEMPO, and ENVELOPE commands in detail. In Part 2, we'll look at the FILTER and SOUND commands and more advanced uses of PLAY.

VOL Means Volume

The 128's VOL command affects all three voices at once and accepts values from 0 (silence) to 15 (maximum). Add the following line to the example program and run it again:

```
10 VOL 15
```

Since the song plays at the same volume, it seems VOL had no effect. In fact, VOL just duplicated the *default* volume setting that PLAY uses when no volume is specified. When you turn on the 128, it establishes several music and sound settings (parameters) in advance. For instance, the PLAY statement above plays the notes at maximum volume with a sound envelope and waveform that simulate a piano. Other default sound parameters, too, remain in effect until you change them.

In many cases you can set the volume at the beginning of a program and leave it alone. However, gradual changes in volume can add to the dynamics of a song. Since drastic volume changes make the SID chip "pop," don't use VOL to turn individual notes on and off. (To hear the pop, turn up the volume on your monitor or TV set, enter the following line without a line number, and press RETURN: VOL 15:VOL 0:VOL 15.)

Unlike PLAY statements, SOUND statements (to be discussed in Part 2) default to a volume of 0. Before using SOUND you must always use VOL to set the volume to some nonzero value.

TEMPO

TEMPO is another command that affects all voices equally, setting the speed at which a song plays. TEMPO is followed by one number in the range 0–255. The default tempo setting is 15, a pedestrian speed. Add the following line to the example program and run it again:

```
20 TEMPO 50
```

At a tempo of 50, the song plays much faster. Try several different TEMPO values in line 20. As you'll find, the highest tempos are exceedingly fast—too speedy for playing whole songs, but handy for simulating trills and grace notes. Change the TEMPO value back to 15 when you're done experimenting with line 20.

Don't confuse tempo—the overall speed of the music—with the individual *duration* of each note (quarter note, sixteenth, etc.). In conventional music a quarter note lasts one "beat," an eighth note lasts one-half beat, and so on. Tempo defines how many beats are played in a minute. At faster tempos every note plays faster, but quarter notes still last twice as long as eighth notes The default note duration for PLAY is a quarter note.

A Built-In Orchestra

The ENVELOPE command is more versatile than VOL or TEMPO. It is used to create customized instrument sounds for your songs. ENVELOPE takes the following general form:

```
ENVELOPE i, a, d, s, r, w, p
```

In the above example, *i* stands for the *instrument number*, *a* for the *attack rate*, *d* for *decay rate*, *s* for *sustain rate*, *r* for *release rate*, *w* for *waveform*, and *p* for *pulsewidth*. Naturally, in a program these letters are replaced with appropriate numbers.

The first number in an ENVELOPE statement chooses one of the 128's *instrument* voices. There are ten predefined instruments, numbered 0–9 as shown here:

Instrument	ENVELOPE #
Piano	0
Accordion	1
Calliope	2
Drum	3
Flute	4
Guitar	5
Harpsichord	6
Organ	7
Trumpet	8
Xylophone	9

Since PLAY commands use the same instrument numbers, you'll want to become familiar with this list. To pick an instrument within PLAY, add a *T* (for *tune*) followed by the desired instrument number. For instance, **PLAY "T5 C D T3 E F"** selects instrument 5 (guitar) and plays notes C and D, then selects instrument 3 (drum) and plays notes E and F. The same numbering scheme identifies customized instruments, as you'll see in a moment. The default instrument for PLAY statements is instrument 0 (piano); if you don't specify an instrument, PLAY always produces a piano sound.

Sound Envelopes

To create new instrument sounds, you'll need to learn about sound envelopes and waveforms. Every natural sound has a distinctive *envelope* or sound pattern. Consider the difference between a snare drum and a violin. Drum sounds begin and end very sharply. The drumhead starts vibrating the instant you strike it, and fades quickly. Violin sounds start out more softly, as the string gradually picks up vibrations from the bow, and fade softly as the vibration dissipates.

The 128 defines different sound envelopes in terms of four values: attack, decay, sustain, and release (ADSR). The *attack* value defines how quickly the sound rises from silence to its peak volume. *Decay* defines how quickly the sound fades from peak volume to the volume at which it will be sustained (held). *Sustain* sets the volume level for the sound's main duration. *Release* defines how quickly the sound fades from its sustained volume back to silence again. Figure 1 illustrates a typical sound envelope.

In ENVELOPE statements, the four numbers after the instrument number define the ADSR envelope. ADSR numbers can range from 0–15.

Waveforms

ENVELOPE also lets you pick different *waveforms*. Each of the SID chip's three voices can produce four different waveforms, diagrammed in Figure 2. The *triangle* waveform (used for the flute, instrument 4) is soft and rich. The *sawtooth* wave (used for the guitar, instrument 5) creates a louder, harsher sound.

The *pulse* waveform (used for the organ, instrument 7) is the most versatile of all. It's louder than the triangle wave and can be adjusted to make sounds that are rich and full or thin and faint. The *noise* waveform (used for the drum, instrument 3) is a random mish-mash of frequencies that make a hissing or rushing sound. ENVELOPE uses the following waveform numbers:

Number	Waveform
0	Triangle
1	Sawtooth
2	Pulse
3	Noise
4	Ring Modulation

Ring modulation is a special effect, different from the other waveforms. The SID chip creates ring modulation by combining the frequencies of two voices into one complex sound. Note that ENVELOPE cannot use *synchronization*, another SID effect familiar to 64 programmers.

Finally, ENVELOPE lets you choose different *pulsewidth* values for the pulse waveform (2). The pulsewidth number can range from 0–4095. Look again at the pulse wave diagram in Figure 2. The top portion of each wave is wider than the bottom portion. The pulsewidth value defines the ratio between these two parts of the wave. Medium pulsewidth values (roughly from 1000–3000) produce fairly symmetrical waves and full, solid tones. Very small or very large pulsewidth values produce assymetrical waves and thin, hollow tones.

ENVELOPE With PLAY

To see what ENVELOPE can do, add line 30 to the example program and insert T1 in line 100 as shown below:

```
30 ENVELOPE 1, 7, 0, 0, 0, 2, 2000
100 PLAY "T1 C D E F G F E D C"
```

Run the program again and notice how different the new instrument sounds. Line 30 selects instrument 1, sets attack at 7, decay, sustain, and release at 0, waveform

at 2 (pulse), and pulsewidth at 2000.

The T1 in line 100 might seem redundant at first: If ENVELOPE selects instrument 1, why specify instrument 1 again in the PLAY statement? This is necessary because of the default system. Until you specify otherwise *in a PLAY statement*, PLAY always uses instrument 0, the piano. Thus, whenever you define a new instrument with ENVELOPE, you must use the *same* instrument number after T in the appropriate PLAY statement. If you forget, PLAY ignores the ENVELOPE statement and uses instrument 0 or whatever instrument you last selected with T.

Redefining an instrument with ENVELOPE replaces the predefined instrument of that number. Thus, you can never have more than ten instruments at once. However, new instruments can be introduced at any time with new ENVELOPE statements.

ENVELOPE can be tricky to handle, since it gives you total control over the ADSR envelope and must be properly integrated with other sound commands. For instance, an envelope that sounds fine at slow tempos may be unsuitable at faster tempos. Don't be discouraged if your first experiments sometimes fail. Remember, ENVELOPE is necessary only for customized instrument sounds. If you're happy with the predefined instruments, just use T in a PLAY statement to choose the one you want.

Figure 1. Typical Sound Envelope

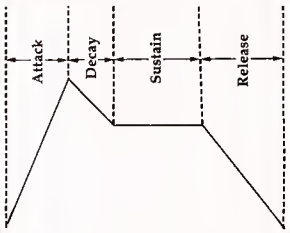
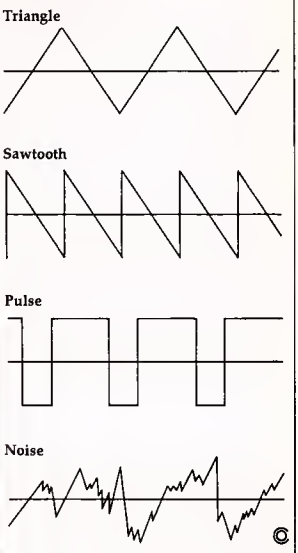


Figure 2. Commodore 128 Waveforms



Colorful Text For IBM Graphics

Peter F. Nicholson, Jr.

IBM personal computers provide a wealth of graphics modes. Here's a method of printing text on graphics screens with different foreground colors to brighten up your screen displays. The technique works on the IBM PCjr with Cartridge BASIC or the PC with BASICA and color/graphics adapter card.

The IBM PCjr and the PC with a color/graphics adapter both have the ability to print different-colored foreground characters on the same colored background. Medium-resolution graphics mode gives you a choice of 16 different background colors and two different foreground color palettes (red/brown/green or cyan/magenta/white). According to page I-9 of the BASIC reference manual for the PC, the following statement changes character colors (substitute 1, 2, or 3 for *color number*):

```
DEF SEG:POKE &H4E,color number
```

Although this changes the character color, it also replaces the background color with color 0

(black). It's not mentioned in the manual, but there is a way to change the foreground color without losing the background color. Simply add 128 to the color number in the above statement. This performs a bit manipulation called an *exclusive OR* (XOR) of the color value, allowing you to print any foreground color on the background.

To see a demonstration, type in and save the program below, then run it. The program illustrates the difference between normal and XOR printing and lets you experiment with many different background and foreground color combinations.

Character Colors For XOR Printing

Color Where Character Will Be XOR PRINTed	Resulting Character Color		
	&H4E= 129	&H4E= 130	&H4E= 131
0	1	2	3
1	0	3	2
2	3	0	1
3	2	1	0

Remember to restore the value in memory location &H4E to either 1, 2, or 3 when you're editing. Otherwise anything you type is XORed with whatever is on the screen. The easiest way to avoid this problem is to clear the screen (press CTRL-HOME) and press function key 10 to edit in SCREEN 0. The table shows the values you can POKE into &H4E to generate various color combinations.

XOR Printing

```
JB 100 SCREEN 1:KEY OFF:PAL=0:BA
       CK=0:GOSUB 1500:COLOR BAC
       K,PAL
BN 110 CLS:COL=3:X0=40:Y0=2B:GOS
       UB 1000:COL=0:GOSUB 1000
FN 120 COL=3:X0=40:Y0=116:GOSUB
       1000:COL=0:GOSUB 1000
GA 130 GOSUB 2000:POKE &H4E,1:LO
       CATE 3,B:PRINT "&H4E=1"
OL 140 FOR I=5 TO 9 STEP 2:LOCAT
       E I,B:PRINT PAL$(PAL,1);:
       NEXT I
KC 150 POKE &H4E,2:LOCATE 3,1B:P
       RINT "&H4E=2"
LN 160 FOR I=5 TO 9 STEP 2:LOCAT
       E I,1B:PRINT PAL$(PAL,2);
       :NEXT I
OI 170 POKE &H4E,3:LOCATE 3,2B:P
       RINT "&H4E=3"
AH 1B0 FOR I=5 TO 9 STEP 2:LOCAT
       E I,2B:PRINT PAL$(PAL,3);
       :NEXT I
NM 190 POKE &H4E,1:LOCATE 14,7:P
       RINT "&H4E=129":POKE &H4E
       ,129
BJ 200 FOR I=16 TO 20 STEP 2:LOC
       ATE I,B:PRINT COL1$(FIX((
       I-16)/2));:NEXT I
BH 210 POKE &H4E,2:LOCATE 14,17:
       PRINT "&H4E=130":POKE &H4
       E,130
LI 220 FOR I=16 TO 20 STEP 2:LOC
       ATE I,1B:PRINT COL2$(FIX(
       (I-16)/2));:NEXT I
LM 230 POKE &H4E,3:LOCATE 14,27:
       PRINT "&H4E=131":POKE &H4
       E,131
AA 240 FOR I=16 TO 20 STEP 2:LOC
       ATE I,2B:PRINT COL3$(FIX(
       (I-16)/2));:NEXT I
LA 250 POKE &H4E,3:LOCATE 1,1:PR
       INT "Normal Printing";
PL 260 LOCATE 12,1:PRINT "XOR Pr
       inting";
KL 270 LOCATE 22,1:PRINT STRING$
       (40,CHR$(32));
DN 2B0 LOCATE 22,1:PRINT "Backgr
       ound ";BACK$(BACK);:LOCAT
       E 22,25:PRINT "Palette ";
       PAL;
BE 290 LOCATE 23,1:PRINT "Press
       Q To Quit"+STRING$(24,CHR
       $(32));
FA 300 LOCATE 24,1:PRINT "Press
       Esc To Change Palette";
II 310 LOCATE 25,1:PRINT "Press
       Space Bar To Change Backg
       round";
JN 320 KB$=INKEY$:IF KB$="" THEN
        330 ELSE 320
ED 330 KB$=INKEY$:IF KB$="" THEN
       330
DB 340 IF KB$="q" OR KB$="Q" THE
       N CLS:END
AG 350 IF ASC(KB$)=32 THEN 3B0 E
       LSE IF ASC(KB$)<27 THEN
       330
NI 360 IF PAL=1 THEN PAL=0 ELSE
       PAL=1
NF 370 COLOR BACK,PAL:GOTO 130
KG 3B0 LOCATE 25,1:PRINT STRING$
       (3B,CHR$(32));
KL 390 LOCATE 24,1:PRINT STRING$
       (3B,CHR$(32));
LL 400 LOCATE 23,1:INPUT "Enter
       Color Number (0-15) ";BAC
       K
MK 410 COLOR BACK,PAL:GOTO 130
OD 1000 PRESET(X0,Y0)
AC 1010 FOR I=1 TO 3
BN 1020 LINE STEP(0,0)-STEP(240,
        16),COL,B
OL 1030 PRESET STEP(-240,0)
GL 1040 NEXT I
EN 1050 PRESET(X0,Y0)
BB 1060 FOR I=1 TO 3
ML 1070 LINE STEP(0,0)-STEP(B0,4
        B),COL,B
AF 10B0 PRESET STEP(0,-4B)
NN 1090 NEXT I:IF COL=0 THEN GOT
        O 1130
HO 1100 FOR I=1 TO 3:FOR J=1 TO
        3
PP 1110 PAINT(X0+20+B0*(I-1),Y0+
        8+16*(J-1)),J,3
IA 1120 NEXT J:NEXT I
IG 1130 RETURN
BN 1500 DIM PAL$(1,3)
KN 1510 FOR I=0 TO 1:FOR J=1 TO
        3:READ PAL$(I,J):NEXT J:
        NEXT I
JB 1520 DATA "GREEN","RED","BROW
        N","CYAN","MAGENTA","WHI
        TE"
NM 1530 DIM BACK$(15)
GG 1540 FOR I=0 TO 15:READ BACK$
        (I):NEXT I
PG 1550 DATA "BLACK","BLUE","GRE
        EN","CYAN","RED","MAGENT
        A","BROWN"
AA 1560 DATA "WHITE","GRAY","L B
        LUE","L GREEN","L CYAN",
        "L RED","L MAGTA"
OI 1570 DATA "YELLOW","HI WHITE"
OL 15B0 BLOCK$=STRING$(B,CHR$(25
        5)):AX=PEEK(VARPTR(BLOCK
        $)+1):BX=PEEK(VARPTR(BLO
        CK$)+2)
NG 1590 DEF SEG=0:POKE 124,AX:PO
        KE 125,BX:POKE 126,PEEK(
        &H510):POKE 127,PEEK(&H5
        11)
NN 1600 DEF SEG:RETURN
EC 2000 COL1$(0)=BACK$(BACK):COL
        1$(1)=PAL$(PAL,3):COL1$(
        2)=PAL$(PAL,2)
CK 2010 COL2$(0)=PAL$(PAL,3):COL
        2$(1)=BACK$(BACK):COL2$(
        2)=PAL$(PAL,1)
EB 2020 COL3$(0)=PAL$(PAL,2):COL
        3$(1)=PAL$(PAL,1):COL3$(
        2)=BACK$(BACK)
IH 2030 DEF SEG:POKE &H4E,1
ME 2040 FOR I=B TO 2B STEP 10:LO
        CATE 5,I:PRINT STRING$(B
        ,CHR$(12B));:NEXT I
KJ 2050 FOR I=B TO 2B STEP 10:LO
        CATE 16,I:PRINT STRING$(
        B,CHR$(12B));:NEXT I
MC 2060 POKE &H4E,2
MD 2070 FOR I=B TO 2B STEP 10:LO
        CATE 7,1:PRINT STRING$(B
        ,CHR$(12B));:NEXT 1
PK 20B0 FOR I=B TO 2B STEP 10:LO
        CATE 1B,I:PRINT STRING$(
        B,CHR$(12B));:NEXT I
OL 2090 POKE &H4E,3
PG 2100 FOR I=B TO 2B STEP 10:LO
        CATE 9,I:PRINT STRING$(B
        ,CHR$(12B));:NEXT I
OK 2110 FOR I=B TO 2B STEP 10:LO
        CATE 20,I:PRINT STRING$(
        B,CHR$(12B));:NEXT I
IE 2120 RETURN
```

Advanced 1541 Disk Commands

Dave Straub

If you want to go beyond the basics of Commodore disk programming, you'll need to learn direct access *disk commands. These powerful commands allow you to read and write individual blocks on a disk. However, since improper use can irretrievably scramble a disk, they are recommended for intermediate and advanced programmers only. You should experiment with them on a scratch disk before attempting to manipulate any important files. The techniques work on any Commodore computer with a 1541 disk drive.*

The 1541 disk drive is a complex device, often called an *intelligent peripheral* because it contains its own microprocessor and operating system. With most computers (such as Apple, Atari, and IBM), the Disk Operating System (DOS) is a program you must load into the computer before using the disk drive. Commodore's DOS, on the other hand, is permanently stored in Read Only Memory (ROM) inside the drive itself.

The Commodore system has some real advantages: DOS does not take up any of the computer's memory, and it's available the instant you turn on the drive. It also makes the 1541 drive independently programmable. By sending *direct access* commands to the drive, you can read or write to any area on the disk, read or write to the drive's internal memory, and even run your own ML programs in the drive.

In this article we'll cover three commands used to manipulate individual disk *blocks*. A block, also called a *sector*, is a small area on the disk that stores 256 bytes of data. As shown in your 1541 *User's Manual*, each disk is divided into 35 separate *tracks*, with each track subdivided into anywhere from 17 to 21 individual blocks or sectors. This yields a total of 683 blocks, each with its own track and block number. For example, the first part of the disk directory is stored in track 18, block 1.

The Command Channel

The first step in most Commodore disk programs is to open the command channel to the drive. This is a special channel used to send instructions to the drive and check for errors. Open it with a line similar to this:

```
10 OPEN 3,8,15
```

This opens communications on channel number 3 to device number 8 (the drive) with a secondary address of 15. The *channel* (sometimes called a *logical file*) number can be anything from 1–15. Once a channel is open, GET#, INPUT#, and PRINT# statements are used *with the channel number* to send or retrieve information on that channel. The number following the # character in these statements must match the channel number used in the OPEN statement. For example, if the channel is opened with OPEN 3,8,15, then PRINT#3,"I0" sends an initialization command to the drive on the command channel.

The *device number* specifies which drive is being accessed. The device number of the 1541 drive is always 8 unless you change it through software or by modifying the drive.

A *secondary address* of 15 has a special meaning: It activates the command channel regardless of what channel number is used. For example, OPEN 1,8,15 or OPEN 15,8,15 both activate the command channel, using channel numbers 1 and 15, respectively.

In this article the command channel is used only to send direct access commands to the drive. However, the command channel also serves the important function of relaying drive error messages to the computer, as explained in your 1541 *User's Manual*.

Buffer Channels

The command channel's abilities are vital but limited: It can only transmit commands and error messages. To transfer *data* (information stored on the disk), you must open a second channel. When this is done, the drive sets aside a 256-byte *buffer* area within its internal memory. It's no coincidence that the buffer is exactly the right size for storing a block of disk data. Since all data moves through the drive's buffers, this type of channel is often called a buffer channel, although data channel might be a more descriptive term. To open a buffer channel, use "#" as a filename in an OPEN command:

```
OPEN 2,8,2,"#"
```

This statement tells the 1541 to open buffer channel 2 to device number 8 with a secondary address of 2, and the special filename "#" reserves a buffer in the drive. The secondary address can be any number from 2–14.

Now that the channel is open, you can find out which buffer the 1541 has reserved. Use GET# to

retrieve the first character available from that channel:

```
10 OPEN 2,8,2,"#":GET#2,X$
20 PRINT "BUFFER"ASC(X$+CHR$
   (0))"USED"
30 CLOSE 2
```

This program opens a data channel and retrieves the number of the buffer reserved for that channel. The 1541 has five 256-byte buffers located at these addresses:

Buffer 0	$0300-03FF	(786-1023)
Buffer 1	$0400-04FF	(1024-1279)
Buffer 2	$0500-05FF	(1280-1535)
Buffer 3	$0600-06FF	(1536-1791)
Buffer 4	$0700-07FF	(1792-2047)

In most cases you needn't worry about which buffer is reserved for your data. The 1541 manages the buffers by itself and always reserves one for you unless none is available. However, by adding a number after the # character, you can force the drive to reserve a specific buffer. For instance, the statement OPEN 2,8,2,"#1" makes the drive set aside buffer number 1 for channel number 2.

To avoid needless errors, don't specify a buffer unless you really need to do so. The buffers are also the 1541's main data area, and at any given time one or more of them may already be in use. For instance, buffer 4 stores the Block Availability Map (BAM) of the current disk and is almost never available. The 1541 generates a NO CHANNEL error message when you try to use a buffer that's already reserved or try to access a channel that wasn't properly opened.

Block-Read

The Block-Read command does exactly what the name implies, reading a block of information from the disk and storing it in a data buffer in the drive. Once the block has been read, you can transfer all or part of it to the computer's memory with GET# or INPUT# statements.

Block-Read has two alternate forms, one that works as expected and another that doesn't. Despite what your *1541 User's Guide* says, don't use the *B-R* form of Block-Read. Use the alternate form (U1) instead. U1 always reads an entire block in correct order, beginning with the first byte of the block and ending with the last. To see how U1 works, type in and save Program 1 below. Program 1 works as listed on the Commodore 64 and the 128 in 64 mode. For the unexpanded VIC, change line 0 to:

```
0 POKE 36879,30:X1=7680:X2=384
  00                        :rem 212
```

For the Commodore 16 or Plus/4, ignore the :rem numbers at the end of each line (they are for the VIC/64 "Automatic Proofreader" program) and change line 0 to:

```
0 X1=3072:X2=2048:COLOR0,2
```

Before running Program 1, *put an unimportant disk in the drive*. Direct access commands are very powerful; even a slight typing error in these programs can garble an entire disk, destroying all of its data. Until you have gained some experience with these commands, it's best to practice on a disk that doesn't contain important programs or data.

When you run Program 1, it displays the 256 bytes stored in track 18, block 0 of your disk. Among other things, this disk block contains the disk name and ID. Line 10 of the program opens the command channel (to send commands to the drive) and line 20 opens the buffer channel to reserve a data buffer. Line 30 contains the Block-Read command (U1). Note that line 30 uses PRINT#3 to send the U1 command. Block-Read is an instruction to the drive; like other instructions it must be sent via the command channel (in this case channel 3).

The actual command is enclosed in quotation marks. First comes the command itself, followed by several parameters separated by spaces. The first number after U1 is 2, telling the drive to read the block into the buffer reserved for channel 2.

The next parameter must always be 0 to indicate drive number 0. (This is a holdover from the old Commodore PET dual drives that are numbered 0 and 1. The 1541 drive is always drive 0, even when two 1541s with different device numbers are daisy-chained together.) The last two parameters inside quotes are the track and sector numbers of the block you wish to read. In this case, 18 and 0 are used to read block 0 of track 18 from the disk.

In line 50 of Program 1, GET#2 retrieves data from the buffer channel. (Since GET# reads incoming zero bytes as null characters—which would crash the ASC function with an error message—it's always necessary to concatenate X$ with CHR$(0).) Line 70 ends the program by closing both channels. Since channel 2 was the last channel opened, it is the first one closed. Always end a program of this type by closing every channel you opened.

Buffer-Pointer

U1 always reads a whole block, but in some cases you'll be interested in only part of the block. For example, you might want one program name from a directory block. The Buffer-Pointer command (abbreviated B-P) is designed for just such occasions. It points the drive to a designated byte within the data buffer, allowing you to read only the bytes you want. The general procedure is to read a block into the buffer with U1, set the pointer with B-P, then retrieve the desired bytes with GET#.

Type in and save Program 2, then run it. The program first reads the block from track 18, block 1 into a buffer. Then B-P sets the buffer pointer to byte 5. Like other direct access commands, B-P is also enclosed in quotes. The second parameter in the command is 2, telling the drive to use buffer channel 2. After 2 comes 5, the number of the byte you want to point at. The remainder of the program simply prints the name of the first program stored on the disk.

Block-Write

The Block-Write command is the opposite of Block-Read, letting you write a block of data from a buffer to any block on the disk. Block-Write also comes in two forms, good and bad. Skip the B-W command; it has the same defects as B-R. Use U2 whenever you want to write a block of data to disk.

To see Block-Write at work, first load Program 2, then add the lines listed below as Program 3. Note that line 80 of Program 3 replaces line 80 of Program 2, and be sure to save a copy of this program

before you run it. This program combines all three of the commands discussed so far. First it reads the name of the first program in the disk directory from block 1 of track 18, then it asks you to type in a new program name. If the new name is less than 16 characters, line 110 "pads" the end of the name with SHIFTed space characters.

You're almost ready to write the new name back to the data buffer. But first it's necessary to set the buffer pointer back to the spot where the old name begins in the buffer (line 120). This step is critical, because GET# affects the buffer pointer much as PRINT affects the cursor on the screen. Every time GET# fetches a character from the buffer, the pointer automatically moves one byte forward. Lines 50–70 repeated GET# 16 times, moving the pointer 16 bytes forward. Thus, before writing the new name back into the buffer, it's necessary to reset the pointer with a second B-P command.

Line 130 writes the new name to the buffer; since the name is data (not an instruction), it's transmitted over the data channel with PRINT#2. Line 140 sends the U2 command over channel 3 to complete the process, telling the drive to write the entire contents of the buffer back to track 18, block 1 of the disk.

As you've probably gathered by now, the 1541 drive handles disk data in block-sized chunks. Though you may want to change only one or two bytes in a block, it's necessary to read the whole block into the drive, make the changes, then write the altered block back to disk again. There's much more to direct access programming, of course, but you can do a great deal with these few commands, moving freely around the disk to examine or modify whatever you find.

Please refer to "COMPUTE!'s Guide to Typing In Programs" in this issue before entering the following listings.

Program 1: Block-Read Demo

```
0 POKE53281,1:X1=1024:X2=55296
                          :rem 146
5 PRINTCHR$(147)CHR$(9)CHR$(14
  ):FORJ=1TO10:PRINT:NEXT
                          :rem 15
10 OPEN3,8,15             :rem 188
20 OPEN2,8,2,"#"          :rem 27
30 PRINT#3,"U1 2 0 18 0"
                          :rem 55
40 FOR X=0 TO 255         :rem 79
50 GET#2,X$:Y=ASC(X$+CHR$(0))
                          :rem 99
60 POKEX1+X,Y:POKEX2+X,0:NEXT
                          :rem 113
70 CLOSE2:CLOSE3          :rem 242
```

Program 2: Buffer-Pointer Demo

```
10 OPEN3,8,15             :rem 188
20 OPEN2,8,2,"#"          :rem 27
30 PRINT#3,"U1 2 0 18 1"
                          :rem 56
40 PRINT#3,"B-P 2 5"      :rem 221
50 FORX=0TO15:GET#2,X$    :rem 49
60 IFX$<>CHR$(160)THENPRINTX$+
   CHR$(0):               :rem 244
70 NEXT                   :rem 166
80 CLOSE2:CLOSE3          :rem 243
```

Program 3: Block-Write Demo

```
80 PRINT:PRINT"WHAT IS THE NEW
    NAME?"                :rem 251
90 INPUT C$               :rem 96
100 FORJ=1TO16:D$=D$+CHR$(160)
    :NEXT                 :rem 14
110 C$=LEFT$(C$+D$,16)    :rem 99
120 PRINT#3,"B-P 2 5"     :rem 12
130 PRINT#2,C$            :rem 9
140 PRINT#3,"U2 2 0 18 1"
                          :rem 107
200 CLOSE2:CLOSE3         :rem 29
```

©

IBM Filecopy

John Klein and Jeff Klein

Here's a fast and easy way to backup multiple files on your disks for safekeeping. It works on any IBM PC, PCjr, or compatible with at least 64K RAM and one or two floppy disk drives.

What is rule number one when you use a computer? *Always make backup copies of all important files.*

But despite one of the most powerful disk operating systems in personal computing, that rule isn't always easy to follow on the IBM. DOS's DISKCOPY utility indiscriminately copies the entire disk, while the COPY command backs up only individual files. Neither allows you to copy groups of specific files from disk to disk or directory to directory very easily. Even if you have two drives, it's time-consuming to combine files from several disks onto a single backup disk, or to backup a group of updated files. As a result, many of us don't make backups as often as we should.

"IBM Filecopy" offers a solution to this problem. It's a utility program that works on any IBM PC, PCjr, or true compatible with one or two disk drives. Filecopy lets

Table 1: Using IBM Filecopy

Type of copy wanted:	Target path to enter:
Same drive, directory "TEST"	\TEST (DOS 2.1 only)
Drive B, same directory	B:
Drive B, directory "TEST"	B:\TEST

Table 2: End of Copy Phase

Screen Message:	Operation/Response:
REM *** COPY COMPLETE	Copies completed.
A>ERASE FILECOPY.BAT	Batch file erases itself.
Insert disk with batch file and press any key when ready	DOS error message. Press any key
A>^C	Press BREAK.
Terminate batch job (Y/N)?	DOS message. Type Y.
A>	Backup process finished; returned to DOS.

you backup disks, directory by directory, or selectively backup individual files. These files can be of any type: BASIC, binary, command, etc. The files can be copied to any subdirectory on any other disk or the same disk.

Using Filecopy

Filecopy is a BASIC program which creates a temporary DOS batch file to copy the specified files to the backup destination. When run, the program first asks you to insert the source disk. This is the disk which contains the files you want to backup. Then the program asks for the source directory of the source disk. If you're not copying from a subdirectory, just type N at this prompt. In either case, Filecopy reads the filenames from the source directory and stores them in an array for later use.

Next the program asks you for the *target path*, the destination for the backups. Type B: for drive B or A: for drive A (be sure to type the colon after the drive designator—B: instead of B). Then type *\directory name* if you're copying the files to a subdirectory on the destination disk. You don't have to specify a directory if you're copying the files to the root (default) directory. If you're using a one-drive system, type B: for the target path as if you really have two drives. Never specify the same drive and directory as the source drive and directory, because the program won't copy files onto themselves. (See Table 1 for sample copy procedures.)

Filecopy then displays each filename from the source directory and asks if you want a copy. Simply type Y for each file you want copied, or N for those you don't want copied. Note: When Filecopy encounters a subdirectory name on your source disk, it's fooled into thinking the subdirectory is a regular file. Since it can't copy subdirectory names, you must answer N when the program asks if you want to backup the subdirectory.

After Filecopy queries you on all of the filenames, it asks for confirmation: "Is this all okay?" If you accidentally typed a wrong Y or N at a previous prompt, type N at this one to get another chance. When you confirm your choices by typing Y, Filecopy stores the names of the files you want copied in a temporary batch file on the source disk. (If the source disk is write-protected, an error message appears and the program halts.) Then it returns you to DOS.

The next step is to type FILECOPY at the DOS prompt. This commands the batch file to copy each of the files you selected from the source disk to the target disk and directory. If you're using a one-drive system, DOS asks you to swap disks as it copies each file.

After the backup is complete, the temporary batch file erases itself off the source disk. This brings up an unavoidable DOS error message, "Insert disk with batch file and press any key when ready." When this message appears, just press CTRL-BREAK on the PC or Fn-B on the PCjr. Another DOS message asks if you want to terminate the batch job. Answer yes. The backup process is now complete. (Table 2 shows the screen messages and responses that should appear during this phase.)

Additional Tips

If you have another BASIC program in memory before running Filecopy, remember to save it on disk. Otherwise it will be replaced when you load and run Filecopy.

Filecopy works with all versions of DOS, but subdirectories are supported only in DOS 2.1 or higher. Do not specify directory changes in the target or source paths if you're using an earlier version of DOS.

If you have two drives and generally use the first drive for the DOS disk and the second for your programming disk, change the first statement in line 100 from P$="A:" to P$="B:". This lets you keep the source disk in drive B and put the target disk in drive A.

IBM Filecopy

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
KG 10 SCREEN 0:WIDTH 80:COLOR 7,
      0
CF 20 CLS:PRINT "Insert source d
      isk in drive A and hit any
       key to continue"
AH 30 GOSUB 700:A$=INPUT$(1)
MM 40 FILES
NE 50 PRINT:INPUT "Enter directo
      ry change (N=None) > ",DIR
      $
CB 60 IF DIR$="N" OR DIR$="n" TH
      EN 100
CH 70 ON ERROR GOTO 720
DK 80 CHDIR DIR$
NP 90 ON ERROR GOTO 0:GOTO 40
DL 100 P$="A:":GOSUB 420:CLS
LF 110 '*** get target path ***
GD 120 INPUT "Target path> ",PAT
      H$
DA 130 IF PATH$="B:" OR PATH$="b
      :" OR PATH$="A:" OR PATH$
      ="a:" OR PATH$="" THEN 19
      0
BK 140 PRINT:INPUT "Do you want
       to create this directory
       on the target disk (Y/N)
       ",A$:IF A$="N" OR A$="n"
       THEN 190 ELSE IF A$<>"Y"
       AND A$<>"y" THEN 140
FI 150 ON ERROR GOTO 730
FB 160 MKDIR PATH$:PRINT:PRINT:P
      RINT
DP 170 '*** get files to copy **
      *
ON 180 ON ERROR GOTO 0
```

```
CK 190 FOR Z=0 TO FILENUM
EE 200 PRINT "COPY>  ";FILE$(Z);
       TAB(24);"?";:COLOR 31,0
HN 210 GOSUB 700
DA 220 A$=INPUT$(1):IF A$<>"Y" A
       ND A$<>"y" AND A$<>"N" AN
       D A$<>"n" THEN 220
LC 230 PRINT SPC(2);A$:COLOR 7,0
OJ 240 IF A$="Y" OR A$="y" THEN
       TY(Z)=1 ELSE TY(Z)=0
IM 250 NEXT Z
GO 260 INPUT "Is this all okay (
       Y/N) ";A$
DG 270 IF A$="N" OR A$="n" THEN
       CLS:GOTO 190
NF 280 IF A$<>"Y" AND A$<>"y" TH
       EN 260
LC 290 '*** batch file creation
       ***
PF 300 OPEN "FILECOPY.BAT" FOR O
       UTPUT AS #1
GJ 310 PRINT #1,"VERIFY ON"
BO 320 FOR Z=0 TO FILENUM
IH 330 IF TY(Z)=0 THEN 350
HO 340 PRINT #1,"COPY ";FILE$(Z)
       ;" ";PATH$
IN 350 NEXT Z
DO 360 PRINT #1,"REM *** COPY CO
       MPLETE"
GG 370 PRINT #1,"ERASE FILECOPY.
       BAT"
AF 380 CLOSE #1
HO 390 '*** enter DOS ***
HN 400 CLS:SYSTEM
AA 410 '*** directory read in ar
       ray FILE$ ***
IK 420 DEF SEG=0
OK 430 CLS:COLOR 31,0:PRINT"One
       moment please"
LA 440 COLOR 7,0:ON ERROR GOTO 4
       60
AD 450 FILES P$:ON ERROR GOTO 0:
       GOTO 470
HN 460 BEEP:COLOR 31:CLS:PRINT"C
       annot read directory":COL
       OR 7:ON ERROR GOTO 0:ENO
BD 470 DIM TEM$(48):LOCATE 3,1:C
       OLOR 7:ROWS=0
LL 480 POKE 1050,30:POKE 1052,34
       :POKE 1054,0:POKE 1055,79
       :POKE 1056,13:POKE 1057,2
       8
EI 490 LINE INPUT TEM$(ROWS)
PS 500 IF TEM$(ROWS)<>"" THEN RO
       WS=ROWS+1:GOTO 480
QE 510 DIM FILE$(ROWS*4-1),TY(RO
       WS*4-1)
HE 520 ROWS=ROWS-1
GJ 530 FOR Z=0 TO ROWS
FC 540 FOR Z1=0 TO 3
DE 550 T$=MID$(TEM$(Z),Z1*18+1,1
       7)
DD 560 IF T$<>"" THEN FILE$(FILE
       NUM)=T$:FILENUM=FILENUM+1
EA 570 NEXT Z1,Z
HB 580 ERASE TEM$:FILENUM=FILENU
       M-1
EN 590 DEF SEG
OF 600 '*** remove spaces from f
       ilename ***
BP 610 FOR Z=0 TO FILENUM
CK 620 A$="":PERIOD=0
CJ 630 FOR Z1=1 TO 17
EA 640 IF MID$(FILE$(Z),Z1,1)="
       " THEN 660 ELSE IF MID$(F
       ILE$(Z),Z1,1)="." THEN PE
       RIOD=1
LN 650 A$=A$+MID$(FILE$(Z),Z1,1)
KN 660 NEXT Z1
EB 670 FILE$(Z)=A$:IF PERIOD=0 T
       HEN FILE$(Z)=FILE$(Z)+"."
EO 680 NEXT Z:RETURN
DD 690 '*** clear keyboard buffe
       r ***
FL 700 DEF SEG=0:POKE 1050,PEEK(
       1052):RETURN
OH 710 '*** error messages ***
BM 720 BEEP:COLOR 31,0:PRINT "Di
       rectory does not exist":C
       OLOR 7,0:RESUME 50
MH 730 BEEP:COLOR 31,0:PRINT:PRI
       NT:PRINT "Cannot create d
       irectory -- reenter path"
       :COLOR 7,0:RESUME 120
```

Apple Text Windows

Daniel L. Joynt

If you use an Apple II-series computer, you may have cast longing eyes at the flashy windowing capabilities of the Macintosh. What you may not realize is that the Apple II has a built-in windowing feature of its own—the text window. The following techniques work on any Apple II-series computer with at least 48K RAM.

While it's difficult for an Apple II-series computer to emulate the slick graphics of the Macintosh, the Apple II does have a simple windowing capability known as the *text window*. Basically, a text window is a section on a low- or high-resolution graphics screen where text can be printed. The text window is easily controlled, too.

You're probably already familiar with some aspects of the Apple II text window. When the computer is in text mode, in effect the text window covers the entire screen—you can print characters wherever you like. In graphics modes, the text window is confined to four lines at the bottom of the screen. When the four-line window fills up, text scrolls off the top of the window to make room for new text at the bottom. Ordinarily the computer manages the text window automatically. However, with a few simple POKEs you can manipulate the text window on your own—controlling its size and screen location.

To see a demonstration of four different text windows, type in, save, and run Program 1. To add windowing to your own BASIC programs, you can use the subroutine in Program 2 (see instructions below).

Open Your Own Windows

As you know, the Apple text screen is 40 columns wide and 24 rows tall. Any screen location can be defined in terms of an X (column) coordinate and a Y (row) coordinate. X coordinates range from 0 (far left) to 39 (far right). Y coordinates range from 0 (top) to 23 (bottom).

The Apple text window is controlled with four memory locations at decimal addresses 32, 33, 34, and 35. By POKEing values in these locations, you can set the boundaries of the window anywhere on the screen. Location 32 defines the left boundary of the text window; it

takes any value from 0–39, corresponding to X coordinate values. Location 33 defines the maximum length of text lines inside the window, which effectively sets the right boundary; it accepts any value from 1–40.

Memory location 34 defines the top boundary of the text window. It takes any value from 0–23, corresponding to Y coordinates. Location 35 defines the window's bottom boundary; it accepts values from 1–24, where 24 is the bottom row of the screen.

To keep a program from crashing, you must keep all the boundaries on the screen. For instance, the values in locations 32 and 33 when added together must not exceed 40. Otherwise, the right boundary would be off the screen. Illogical configurations—such as setting the top boundary below the bottom boundary—also cause a crash.

Give The Cursor A Home

Once the window is opened, you'll want to fill it with text. Unfortunately, resetting locations 32–35 does not automatically relocate the cursor inside the new window. The HOME command sends the cursor to the upper-left corner, but also erases everything inside the window. To move the cursor less destructively, you can use the HTAB and VTAB commands.

HTAB and VTAB set the screen location where all following text will be printed. HTAB moves the text pointer to a specified column (1–40), and VTAB moves the text pointer to the designated row (1–24). Note that HTAB and VTAB act differently when used outside of a text window.

An Easy Window Routine

Program 2 is a subroutine that creates a text window of any size and shape and even surrounds it with a border if you like. It uses line numbers 10000–10070, but you can renumber the lines when adding the routine to your own programs. Note that this is a subroutine, not a complete program; certain steps must be followed when using the routine *after* it has been added to another program.

Before calling the subroutine with GOSUB 10000, you must give each of the following variables a value within the range indicated. In addition to setting these variables, your program should not use the variables WX and WY, which the routine uses to draw borders.

Variable	Range
WL	2–39
WR	2–39
WT	2–22
WB	2–22
W$	any single character
WI	0–2

Line 10000 resets the text window to the size of the entire screen. This insures that HTAB and VTAB place the border properly. In lines 10010–10030 the variable WI defines the print mode of the border. A value of 0 maintains NORMAL mode, 1 sets the mode to INVERSE, and 2 sets it to FLASH.

The variable W$ defines the character used for the border, which is drawn in lines 10040–10050. If you don't want a border, define W$ as a null string (W$=""). Line 10060 locates the new text window, setting the left, right, top, and bottom boundaries with the variables WL, WR, WT, and WB, respectively. Note that the left and right boundaries (WL and WR) must be in the range 2–39 to allow room for a border. For the same reason, WT and WB (top and bottom boundaries) should be in the range 2–22. Once the window is defined, line 10070 resets the Apple to NORMAL mode, clears the inside of the new text window, and ends the routine.

Program 1: Window Demonstration

```
29 100 TEXT
46 110 HOME
F1 120 REM -- DRAW MENU WINDDW
F7 130 WL = 10:WR = 30:WT = 5:WB
     = 18:W$ = "*":WI = 0
DF 140 GDSUB 10000
EA 150 PRINT
F8 160 PRINT TAB( 4);"==== MENU
     ===="
6C 170 PRINT : PRINT
89 180 FOR Y = 1 TO 4
34 190 PRINT TAB( 4);Y;"- WINDDW
     (";CHR$ (64 + Y);")"
E6 200 NEXT Y
DF 210 PRINT : PRINT : PRINT
C5 220 PRINT "CHDOSE WINDOW (1-4
     )";
DA 230 GET Y$
DD 240 IF Y$ < "1" DR Y$ > "4" T
     HEN GOTD 100
69 250 W = VAL (Y$)
B4 260 ON W GOSUB 1000,2000,3000
     ,4000
EC 270 FDR Z = 1 TD 1000: NEXT Z
99 280 LIST
57 290 HOME
BF 300 PRINT : PRINT : PRINT
CA 310 PRINT TAB( 3);"WINDOW (";
     CHR$ (64 + W);")"
E3 320 FOR Z = 1 TO 1000: NEXT Z
86 330 REM -- DRAW RETURN WINDOW
88 340 WL = 6:WR = 34:WT = 12:WB
     = 12:W$ = CHR$ (32):WI =
     1
D4 350 GOSUB 10000
62 360 PRINT TAB( 5);"PLEASE PRE
     SS ANY KEY";
E3 370 GET Y$
9F 380 GOTO 130
C8 1000 REM -- DRAW WINDOW (A)
FA 1010 WL = 2:WR = 15:WT = 2:WB
     = 8:W$ = CHR$ (32):WI =
     1
A9 1020 GOSUB 10000
DD 1030 RETURN
D9 2000 REM -- DRAW WINDOW (B)
3E 2010 WL = 26:WR = 39:WT = 2:W
     B = 8:W$ = "*":WI = 1
4A 2020 GOSUB 10000
DE 2030 RETURN
EA 3000 REM -- DRAW WINDDW (C)
D8 3010 WL = 2:WR = 15:WT = 16:W
     B = 22:W$ = "@":WI = 0
48 3020 GOSUB 10000
DF 3030 RETURN
FB 4000 REM -- DRAW WINDOW (D)
94 4010 WL = 26:WR = 39:WT = 16:
     WB = 22:W$ = "+":WI = 1
4C 4020 GDSUB 10000
E8 4030 RETURN
B4 10000 POKE 32,0: PDKE 33,40:
     POKE 34,0: POKE 35,24
DB 10010 IF WI = 0 THEN NORMAL
FD 10020 IF WI = 1 THEN INVERSE
4E 10030 IF WI = 2 THEN FLASH
89 10040 FOR WY = WT - 1 TO WB +
     1: VTAB WY: HTAB WL -
     1: PRINT W$;: HTAB WR +
     1: PRINT W$;: NEXT
C9 10050 FDR WX = WL TO WR: VTAB
     WT - 1: HTAB WX: PRINT
     W$;: VTAB WB + 1: HTAB
     WX: PRINT W$;: NEXT
6F 10060 POKE 32,WL - 1: POKE 33
     ,WR - WL + 1: POKE 34,W
     T - 1: POKE 35,WB
96 10070 NORMAL : HOME : RETURN
```

Program 2: Window Subroutine

```
B4 10000 PDKE 32,0: POKE 33,40:
     POKE 34,0: POKE 35,24
DB 10010 IF WI = 0 THEN NORMAL
FD 10020 IF WI = 1 THEN INVERSE
4E 10030 IF WI = 2 THEN FLASH
89 10040 FDR WY = WT - 1 TO WB +
     1: VTAB WY: HTAB WL -
     1: PRINT W$;: HTAB WR +
     1: PRINT W$;: NEXT
C9 10050 FOR WX = WL TO WR: VTAB
     WT - 1: HTAB WX: PRINT
     W$;: VTAB WB + 1: HTAB
     WX: PRINT W$;: NEXT
6F 10060 POKE 32,WL - 1: PDKE 33
     ,WR - WL + 1: PDKE 34,W
     T - 1: POKE 35,WB
96 10070 NDRMAL : HOME : RETURN
```

COMPUTE!'s Guide To Typing In Programs

Before typing in any program, you should familiarize yourself with your computer. Learn how to use the keyboard to type in and correct BASIC programs. Read your manuals to understand how to save and load BASIC programs to and from your disk drive or cassette unit. Computers are precise—take special care to type the program *exactly* as listed, including any necessary punctuation and symbols, except for special characters as noted below. To help you with this task, we have implemented a special listing convention as well as a program to help check your typing—the "Automatic Proofreader." Please read the following notes before typing in any programs from COMPUTE!. They can save you a lot of time and trouble.

Commodore, Apple, and Atari programs can contain some hard-to-read (and hard-to-type) special characters, so we have developed a listing system that indicates the function of these control characters. (There are no special control characters in our IBM or TI-99/4A listings.) You will find Commodore and Atari special characters within curly braces; *do not type the braces.* For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. For Commodore, Apple, and Atari, a symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CTRL key and press A. This will produce a reverse video character on the Commodore (in quote mode), a graphics character on the Atari, and an invisible control character on the Apple. Commodore computers also have a special control key labeled with the Commodore logo. Graphics characters entered with the Commodore logo key are enclosed in a special bracket that looks like this: [<A>]. In this case, you would hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S. One exception is {SHIFT-SPACE}. When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, such as {5 RIGHT}, {6 S}, or [<8 Q>], you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (printed in white on black) should be entered after pressing the inverse video key.

Since spacing is sometimes important, any more than two spaces will be listed. For example, {6 SPACES} means to press the space bar six times. Our listings never leave a space at the end of a line, instead moving it to the next printed line as {SPACE}. For your convenience, we have prepared this quick-reference chart for the Commodore and Atari special characters:

Atari 400/800/XL/XE

When you see	Type	See	
{CLEAR}	ESC SHIFT <		Clear Screen
{UP}	ESC CTRL -		Cursor Up
{DOWN}	ESC CTRL =		Cursor Down
{LEFT}	ESC CTRL +		Cursor Left
{RIGHT}	ESC CTRL *		Cursor Right
{BACK S}	ESC DELETE		Backspace
{DELETE}	ESC CTRL DELETE		Delete character
{INSERT}	ESC CTRL INSERT		Insert character
{DEL LINE}	ESC SHIFT DELETE		Delete line
{INS LINE}	ESC SHIFT INSERT		Insert line
{TAB}	ESC TAB		TAB key
{CLR TAB}	ESC CTRL TAB		Clear tab
{SET TAB}	ESC SHIFT TAB		Set tab stop
{BELL}	ESC CTRL 2		Ring buzzer
{ESC}	ESC ESC		ESCape key

Commodore PET/CBM/VIC 64/128/16/+4

When You Read:	Press:	See:	When You Read:	Press:	See:
{CLR}	SHIFT + CLR/HOME		[<1>]	COMMODORE + 1	
{HOME}	CLR/HOME		[<2>]	COMMODORE + 2	
{UP}	SHIFT + ↑ CRSR ↓		[<3>]	COMMODORE + 3	
{DOWN}	↑ CRSR ↓		[<4>]	COMMODORE + 4	
{LEFT}	SHIFT + ← CRSR →		[<5>]	COMMODORE + 5	
{RIGHT}	← CRSR →		[<6>]	COMMODORE + 6	
{RVS}	CTRL + 9		[<7>]	COMMODORE + 7	
{OFF}	CTRL + 0		[<8>]	COMMODORE + 8	
{BLK}	CTRL + 1		{ F1 }	f1	
{WHT}	CTRL + 2		{ F2 }	SHIFT + f1	
{RED}	CTRL + 3		{ F3 }	f3	
{CYN}	CTRL + 4		{ F4 }	SHIFT + f3	
{PUR}	CTRL + 5		{ F5 }	f5	
{GRN}	CTRL + 6		{ F6 }	SHIFT + f5	
{BLU}	CTRL + 7		{ F7 }	f7	
{YEL}	CTRL + 8		{ F8 }	SHIFT + f7	
			4	←	

The Automatic Proofreader

We have developed a series of simple, yet effective programs that can help check your typing. Type in the appropriate Proofreader program listed below, then save it for future use. On the VIC, 64, or Atari, run the Proofreader to activate it, then enter NEW to erase the BASIC loader (the Proofreader remains active, hidden in memory, as a machine language program). Pressing RUN/STOP–RESTORE or SYSTEM RESET deactivates the Proofreader. You can use SYS 886 to reactivate the VIC/64 Proofreader, or PRINT USR(1536) to reenable the Atari Proofreader. On the Apple, the Proofreader automatically erases the BASIC portion of itself after you activate it by typing RUN, leaving only the machine language portion in memory. It works with either DOS 3.3 or ProDOS. Disable the Apple Proofreader by pressing CTRL-RESET before running another BASIC program. The IBM Proofreader is a BASIC program that simulates the IBM BASIC line editor, letting you enter, edit, list, save, and load programs that you type. Type RUN to activate.

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a decimal number (on the Commodore), a hexadecimal number (on the Apple), or a pair of letters (on the Atari or IBM) appears. The number or pair of letters is called a *checksum*. Try making a change in the line, and notice how the checksum changes.

All you need to do is compare the value provided by the Proofreader with the checksum printed in the program listing in the magazine. In Commodore listings, the checksum is a number from 0 to 255. It is set off from the rest of the line with *rem*. This prevents a syntax error if the checksum is typed in, but the REM statements and checksums need *not* be typed in. It is just there for your information.

In Atari, Apple, and IBM listings, the checksum is given to the left of each line number. Just type in the program one line at a time (without the printed checksum) and compare the checksum generated by the Proofreader to the checksum in the listing. If they match, go on to the next line. If not, check your typing: You've made a mistake. On the Commodore, Atari, and Apple Proofreaders, spaces are not counted as part of the checksum, so be sure you type the right number of spaces between quote marks. The Commodore and Atari Proofreaders do not check to see that you've typed the characters in the right order, so if characters are transposed, the checksum still matches the listing. Because of the checksum method used, do not type abbreviations, such as ? for PRINT. The IBM Proofreader is the pickiest of all; it *will* detect errors in spacing and transposition. Be sure to leave Caps Lock on, except when typing lowercase characters.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you type NEW, the Proofreader prompts you to press Y to be sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program in BASIC as usual (this replaces the Proofreader in memory). You can now run the program, but you may want to resave it to disk. The version of your program that you resave from BASIC will take up less space on disk and will load faster, but it can no longer be edited with the Proofreader. If you want to convert a program to Proofreader format, save it to disk with SAVE *"filename"*,A.

Special Proofreader Notes For Commodore Cassette Users

The Proofreader resides in a section of memory called the cassette buffer, which is used during tape LOADs and SAVEs. Therefore, be sure to press RUN/STOP–RESTORE to get the Proofreader out of the way before saving or loading a program. If you want to use the Proofreader with tape, run the Proofreader, then enter these two lines *exactly* as shown, pressing RETURN after each one:

```
A$="PROOFREADER.T":B$="{10
 SPACES}":FOR X=1 TO 4:A$=A$
 +B$:NEXT
FOR X=886 TO 1018:A$=A$+CHR$
(PEEK(X)):NEXT:OPEN 1,1,1,A$:
 CLOSE1
```

Then insert a blank tape and press RECORD and PLAY to save a special version of the Proofreader. Anytime you need to reload the Proofreader after it has been erased—for example, after you reload a paritally completed program—just rewind the tape, type OPEN1:CLOSE1, then press PLAY. You'll see the message FOUND PROOFREADER.T, but not the familiar LOADING message. Don't worry; the Proofreader is in memory. When READY comes back, enter SYS 886.

Program 1: VIC/64 Proofreader

By Charles Brannon, Program Editor

```
10 PRINT"{CLR}PLEASE WAIT...":
   FORI=886TO1018:READA:CK=CK+
   A:POKEI,A:NEXT
20 IF CK<>17539 THEN PRINT"
   {DOWN}YOU MADE AN ERROR":PR
   INT"IN DATA STATEMENTS.":EN
   D
30 SYS886:PRINT"{CLR}{2 DOWN}P
   ROOFREADER ACTIVATED.":NEW
40 DATA 173,036,003,201,150,20
   8,001,096,141,151,003,173
50 DATA 037,003,141,152,003,16
   9,150,141,036,003,169,003
60 DATA 141,037,003,169,000,13
   3,254,096,032,087,241,133
70 DATA 251,134,252,132,253,00
   8,201,013,240,017,201,032
80 DATA 240,005,024,101,254,13
   3,254,165,251,166,252,164
90 DATA 253,040,096,169,013,03
   2,210,255,165,214,141,251
100 DATA 003,206,251,003,169,0
   00,133,216,169,019,032,210
110 DATA 255,169,018,032,210,2
   55,169,58,032,210,255,166
120 DATA 254,169,000,133,254,1
   72,151,003,192,087,208,006
130 DATA 032,205,189,076,235,0
   03,032,205,221,169,032,032
140 DATA 210,255,032,210,255,1
   73,251,003,133,214,076,173
150 DATA 003
```

Program 2: Atari Proofreader

By Charles Brannon, Program Editor

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:RE
    AD A:POKE I,A:CK=CK+A
    :NEXT I
120 IF CK<>19072 THEN ? "
    Error in DATA Stateme
    nts.  Check Typing.":
    END
130 A=USR(1536)
140 ? :? "Automatic Proof
    reader Now Activated.
    "
150 END
160 DATA 104,160,0,185,26
    ,3,201,69,240,7
170 DATA 200,200,192,34,2
    08,243,96,200,169,74
180 DATA 153,26,3,200,169
    ,6,153,26,3,162
190 DATA 0,189,0,228,157,
    74,6,232,224,16
200 DATA 208,245,169,93,1
    41,78,6,169,6,141
210 DATA 79,6,24,173,4,22
    8,105,1,141,95
```

```
220 DATA 6,173,5,228,105,
    0.141,96,6,169
230 DATA 0,133,203,96,247
    .238,125,241,93,6
240 DATA 244,241,115,241,
    124,241,76,205,238
250 DATA 0,0,0,0,0,0,32,62,
    246,8,201
260 DATA 155,240,13,201,3
    2,240,7,72,24,101
270 DATA 203,133,203,104,
    40,96,72,152,72,138
280 DATA 72,160,0,169,128
    ,145,88,200,192,40
290 DATA 208,249,165,203,
    74,74,74,74,24,105
300 DATA 161,160,3,145,88
    ,165,203,41,15,24
310 DATA 105,161,200,145,
    88,169,0,133,203,104
320 DATA 170,104,168,104,
    40,96
```

Program 3: IBM Proofreader

By Charles Brannon, Program Editor

```
10 'Automatic Proofreader Ver
   sion 2.00 (Lines 270,510,5
   15,517,620,630 changed fro
   m V1.0)
100 DIM L$(500),LNUM(500):COL
   OR 0,7,7:KEY OFF:CLS:MAX=
   0:LNUM(0)=65536!
110 ON ERRDR GDTD 120:KEY 15,
   CHR$(4)+CHR$(70):ON KEY(1
   5) GOSUB 640:KEY (15) ON:
   GDTO 130
120 RESUME 130
130 OEF SEG=&H40:W=PEEK(&H4A)
140 ON ERROR GOTO 650:PRINT:P
   RINT"Proofreader Ready."
150 LINE INPUT L$:Y=CSRLIN-IN
   T(LEN(L$)/W)-1:LOCATE Y,1
160 OEF SEG=0:POKE 1050,30:PO
   KE 1052,34:POKE 1054,0:PO
   KE 1055,79:POKE 1056,13:P
   OKE 1057,28:LINE INPUT L$
   :OEF SEG:IF L$="" THEN 15
   0
170 IF LEFT$(L$,1)=" " THEN L
   $=MID$(L$,2):GOTD 170
180 IF VAL(LEFT$(L$,2))=0 ANO
    MID$(L$,3,1)=" " THEN L$
   =MID$(L$,4)
190 LNUM=VAL(L$):TEXT$=MID$(L
   $,LEN(STR$(LNUM))+1)
200 IF ASC(L$)>57 THEN 260 'n
   o Iine number, therefore
   command
210 IF TEXT$="" THEN GOSUB 54
   0:IF LNUM=LNUM(P) THEN GO
   SUB 560:GOTO 150 ELSE 150
220 CKSUM=0:FOR I=1 TO LEN(L$
   ):CKSUM=(CKSUM+ASC(MID$(L
   $,I))*I) ANO 255:NEXT:LOC
   ATE Y,1:PRINT CHR$(65+CKS
   UM/16)+CHR$(65+(CKSUM ANO
    15))+" "+L$
230 GOSUB 540:IF LNUM(P)=LNUM
    THEN L$(P)=TEXT$:GOTO 15
   0 'replace Iine
240 GOSUB 580:GOTO 150 'inser
   t the Iine
260 TEXT$="":FOR I=1 TO LEN(L
   $):A=ASC(MIO$(L$,I)):TEXT
   $=TEXT$+CHR$(A+32*(A>96 A
   NO A<123)):NEXT
270 DELIMITER=INSTR(TEXT$," "
   ):CDMMAND$=TEXT$:ARG$="":
    IF DELIMITER THEN CDMMAND
   $=LEFT$(TEXT$,DELIMITER-1
   ):ARG$=MID$(TEXT$,DELIMIT
   ER+1) ELSE DELIMITER=INST
   R(TEXT$,CHR$(34)):IF DELI
   MITER THEN COMMAND$=LEFT$
   (TEXT$,OELIMITER-1):ARG$=
   MID$(TEXT$,DELIMITER)
280 IF CDMMAND$<>"LIST" THEN
    410
290 DPEN "scrn:" FDR DUTPUT A
   S #1
300 IF ARG$="" THEN FIRST=0:P
   =MAX-1:GDTO 340
310 DELIMITER=INSTR(ARG$,"-")
   :IF DELIMITER=0 THEN LNUM
   =VAL(ARG$):GDSUB 540:FIRS
   T=P:GDTO 340
320 FIRST=VAL(LEFT$(ARG$,OELI
   MITER)):LAST=VAL(MID$(ARG
   $,OELIMITER+1))
330 LNUM=FIRST:GDSUB 540:FIRS
   T=P:LNUM=LAST:GDSUB 540:I
   F P=0 THEN P=MAX-1
340 FOR X=FIRST TO P:N$=MID$(
   STR$(LNUM(X)),2)+" "
350 IF CKFLAG=0 THEN A$="":GD
   TO 370
360 CKSUM=0:A$=N$+L$(X):FDR I
   =1 TD LEN(A$):CKSUM=(CKSU
   M+ASC(MID$(A$,I))*I) AND
   255:NEXT:A$=CHR$(65+CKSUM
   /16)+CHR$(65+(CKSUM AND 1
   5))+" "
370 PRINT #1,A$+N$+L$(X)
380 IF INKEY$<>"" THEN X=P
390 NEXT :CLDSE #1:CKFLAG=0
400 GDTD 130
410 IF CDMMAND$="LLIST" THEN
   DPEN "lpt1:" FOR OUTPUT A
   S #1:GOTO 300
420 IF COMMAND$="CHECK" THEN
   CKFLAG=1:GOTO 290
430 IF COMMAND$<>"SAVE" THEN
   450
440 GOSUB 600:OPEN ARG$ FOR O
   UTPUT AS #1:ARG$="":GOTO
   300
450 IF CDMMAND$<>"LOAO" THEN
   490
460 GOSUB 600:OPEN ARG$ FOR I
   NPUT AS #1:MAX=0:P=0
470 WHILE NOT EOF(1):LINE INP
   UT #1,L$:LNUM(P)=VAL(L$):
   L$(P)=MIO$(L$,LEN(STR$(VA
   L(L$)))+1):P=P+1:WENO
480 MAX=P:CLOSE #1:GOTO 130
490 IF COMMAND$="NEW" THEN IN
   PUT "Erase program - Are
   you sure";L$:IF LEFT$(L$,
   1)="y" OR LEFT$(L$,1)="Y"
    THEN MAX=0:GOTD 130:ELSE
    130
500 IF COMMANO$="BASIC" THEN
   COLOR 7,0,0:ON ERROR GOTO
    0:CLS:ENO
510 IF CDMMAND$<>"FILES" THEN
    520
515 IF ARG$="" THEN ARG$="A:"
    ELSE SEL=1:GOSUB 600
517 FILES ARG$:GOTO 130
520 PRINT"Syntax error":GOTO
    130
540 P=0:WHILE LNUM>LNUM(P) AN
   O P<MAX:P=P+1:WEND:RETURN
560 MAX=MAX-1:FOR X=P TO MAX:
   LNUM(X)=LNUM(X+1):L$(X)=L
   $(X+1):NEXT:RETURN
580 MAX=MAX+1:FDR X=MAX TD P+
   1 STEP -1:LNUM(X)=LNUM(X-
   1):L$(X)=L$(X-1):NEXT:L$(
   P)=TEXT$:LNUM(P)=LNUM:RET
   URN
600 IF LEFT$(ARG$,1)<>CHR$(34
   ) THEN 520 ELSE ARG$=MID$
   (ARG$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34
   ) THEN ARG$=LEFT$(ARG$,LE
   N(ARG$)-1)
620 IF SEL=0 AND INSTR(ARG$,"
   .")=0 THEN ARG$=ARG$+".BA
   S"
630 SEL=0:RETURN
640 CLDSE #1:CKFLAG=0:PRINT"S
   topped.":RETURN 150
650 PRINT "Error #";ERR:RESUM
   E 150
```

Program 4: Apple Proofreader

By Tim Victor, Editorial Programmer

```
10 C = 0: FOR I = 768 TO 768 +
   68: READ A:C = C + A: PDKE I
   ,A: NEXT
20 IF C < > 7258 THEN PRINT "ER
   ROR IN PRDDFREADER OATA STAT
   EMENTS": END
30 IF PEEK (190 * 256) < > 76 T
   HEN PDKE 56,0: POKE 57,3: CA
   LL 1002: GOTO 50
40 PRINT CHR$ (4);"IN#A$300"
50 POKE 34,0: HDME : POKE 34,1:
    VTAB 2: PRINT "PROOFREADER
   INSTALLEO"
60 NEW
100 DATA 216,32,27,253,201,141
110 DATA 208,60,138,72,169,0
120 OATA 72,189,255,1,201,160
130 OATA 240,8,104,10,125,255
140 OATA 1,105,0,72,202,208
150 OATA 238,104,170,41,15,9
160 DATA 48,201,58,144,2,233
170 DATA 57,141,1,4,138,74
180 DATA 74,74,74,41,15,9
190 OATA 48,201,58,144,2,233
200 OATA 57,141,0,4,104,170
210 OATA 169,141,96
```

MLX Machine Language Entry Program For Commodore 64

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost fail-safe entry of machine language programs published in COMPUTE!. You need to know nothing about machine language to use MLX—it was designed for everyone. At least 8K expansion memory is required.

MLX is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file.

Using MLX

Type in and save the appropriate version of MLX (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. MLX for the 64 asks you for two numbers: the starting address and the ending address. These numbers are given in the article accompanying the ML program.

When you run MLX, you'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a *checksum number*. The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. If you make a typing error, press the INST/DEL key; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer automatically prints the comma and goes on to accept the next number. If you enter less than three digits, you can press either the space bar or RETURN key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

To simplify your typing, MLX redefines part of the keyboard as a numeric keypad (lines 581–584):

```
    U  I  O                7  8  9
H   J  K  L  become  0     4  5  6
    M  ,  .                1  2  3
```

64 MLX Commands

When you finish typing an ML listing (assuming you type it all in one session), you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. MLX recognizes these commands:

SHIFT-S: Save
SHIFT-L: Load
SHIFT-N: New Address
SHIFT-D: Display

When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command to save what you've been working on. It will save on tape or disk, as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember what address you stop at. The next time you run MLX, answer all the prompts as you did before, then insert the disk or tape. When you get to the entry prompt, press SHIFT-L to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press SHIFT-N and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the special listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press SHIFT-D, enter two addresses within the line number range of the listing. You can abort the listing by pressing any key.

64 MLX: Machine Language Entry

```
10 REM LINES CHANGED FROM MLX
   {SPACE}VERSION 2.00 ARE 750
   ,765,770 AND 860      :rem 50
20 REM LINE CHANGED FROM MLX V
   ERSION 2.01 IS 300 :rem 147
100 PRINT"{CLR}{63}";CHR$(142);
    CHR$(8);:POKE53281,1:POKE5
    3280,1                :rem 67
101 POKE 788,52:REM DISABLE RU
    N/STOP                :rem 119
110 PRINT"{RVS}{39 SPACES}";
                          :rem 176
120 PRINT"{RVS}{14 SPACES}
    {RIGHT}{OFF}{*}{£}{RVS}
    {RIGHT} {RIGHT}{2 SPACES}
    {*}{OFF}{*}{£}{RVS}{£}{RVS}
    {14 SPACES}";         :rem 250
130 PRINT"{RVS}{14 SPACES}
    {RIGHT} {G}{RIGHT}
    {2 RIGHT} {OFF}{£}{RVS}{£
    {*}{OFF}{*}{RVS}
    {14 SPACES}";         :rem 35
140 PRINT"{RVS}{41 SPACES}"
                          :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK} M
    ACHINE LANGUAGE EDITOR VER
    SION 2.02{5 DOWN}":rem 238
210 PRINT"{5}{2 UP}STARTING AD
    DRESS?{8 SPACES}{9 LEFT}";
                          :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+11
    9*F)                  :rem 166
220 IFS<256ORS>40960ANDS<4915
    2)ORS>53247THENGOSUB3000:G
    OTO210                :rem 235
225 PRINT:PRINT:PRINT :rem 180
230 PRINT"{5}{2 UP}ENDING ADDR
    ESS?{8 SPACES}{9 LEFT}";:I
    NPUTE:F=1-F:C$=CHR$(31+119
    *F)                   :rem 20
240 IFE<256ORE>40960ANDE<4915
    2)ORE>53247THENGOSUB3000:G
    OTO230                :rem 183
250 IFE<STHENPRINTC$;"{RVS}END
    ING < START{2 SPACES}":GOS
    UB1000:GOTO 230       :rem 176
260 PRINT:PRINT:PRINT :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S
                          :rem 56
310 A=1:PRINTRIGHT$("0000"+MID
    $(STR$(AD),2),5);":";
                          :rem 33
315 FORJ=ATO6             :rem 33
320 GOSUB570:IFN=-1THENJ=J+N:G
    OTO320                :rem 228
390 IFN=-211THEN 710      :rem 62
400 IFN=-204THEN 790      :rem 64
410 IFN=-206THENPRINT:INPUT"
    {DOWN}ENTER NEW ADDRESS";Z
    Z                     :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETH
    ENPRINT"{RVS}OUT OF RANGE"
    :GOSUB1000:GOTO410:rem 225
417 IFN=-206THENAD=ZZ:PRINT:GO
    TO310                 :rem 238
420 IF N<>-196 THEN 480
                          :rem 133
430 PRINT:INPUT"DISPLAY FROM";
    F:PRINT,"TO";:INPUTT
                          :rem 234
440 IFF<SORF>EORT<SORT>ETHENPR
    INT"AT LEAST";S;"{LEFT}, N
    OT MORE THAN";E:GOTO430
                          :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINT
    RIGHT$("0000"+MID$(STR$(I)
    ,2),5);":";           :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRIN
    TRIGHT$("00"+MID$(STR$(N),
    2),3);",";            :rem 66
```

```
460 GETA$:IFA$>""THENPRINT:PRI
    NT:GOTO310          :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI
    :PRINT:PRINT:GOTO310
                        :rem 50
480 IFN<0 THEN PRINT:GOTO310
                        :rem 168
490 A(J)=N:NEXTJ        :rem 199
500 CKSUM=AD-INT(AD/256)*256:F
    ORI=1TO6:CKSUM=(CKSUM+A(I)
    )AND255:NEXT        :rem 200
510 PRINTCHR$(18);:GOSUB570:PR
    INTCHR$(146);       :rem 94
511 IFN=-1THENA=6:GOTO315
                        :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHE
    N530                :rem 122
520 PRINT:PRINT"LINE ENTERED W
    RONG : RE-ENTER":PRINT:GOS
    UB1000:GOTO310      :rem 176
530 GOSUB2000           :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):
    NEXT:POKE54272,0:POKE54273
    ,0                  :rem 227
550 AD=AD+6:IF AD<E THEN 310
                        :rem 212
560 GOTO 710            :rem 108
570 N=0:Z=0             :rem 88
580 PRINT"{£3}";        :rem 81
581 GETA$:IFA$=""THEN581
                        :rem 95
582 AV=-(A$="M")-2*(A$=",")-3*
    (A$=".")-4*(A$="J")-5*(A$=
    "K")-6*(A$="L")     :rem 41

583 AV=AV-7*(A$="U")-8*(A$="I"
    )-9*(A$="O"):IFA$="H"THENA
    $="0"               :rem 134
584 IFAV>0THENA$=CHR$(48+AV)
                        :rem 134
585 PRINTCHR$(20);:A=ASC(A$):I
    FA=13ORA=44ORA=32THEN670
                        :rem 229
590 IFA>128THENN=-A:RETURN
                        :rem 137
600 IFA<>20 THEN 630    :rem 10
610 GOSUB690:IFI=1ANDT=44THENN
    =-1:PRINT"{OFF}{LEFT}
    {LEFT}";:GOTO690    :rem 62
620 GOTO570             :rem 109
630 IFA<48ORA>57THEN580
                        :rem 105
640 PRINTA$;:N=N*10+A-48
                        :rem 106
650 IFN>255 THEN A=20:GOSUB100
    0:GOTO600           :rem 229
660 Z=Z+1:IFZ<3THEN580 :rem 71
670 IFZ=0THENGOSUB1000:GOTO570
                        :rem 114
680 PRINT",";:RETURN    :rem 240
690 S%=PEEK(209)+256*PEEK(210)
    +PEEK(211)          :rem 149
691 FORI=1TO3:T=PEEK(S%-I)
                        :rem 67
695 IFT<>44ANDT<>58THENPOKES%-
    I,32:NEXT           :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1)
    ;:RETURN            :rem 7
710 PRINT"{CLR}{RVS}*** SAVE *
    **{3 DOWN}"         :rem 236
715 PRINT"{2 DOWN}{PRESS {RVS}
    RETURN{OFF} ALONE TO CANCE
    L SAVE}{DOWN}"      :rem 106
720 F$="":INPUT"{DOWN} FILENAM
    E";F$:IFF$=""THENPRINT:PRI
    NT:GOTO310          :rem 71
730 PRINT:PRINT"{2 DOWN}{RVS}T
    {OFF}APE OR {RVS}D{OFF}ISK
    : (T/D)"            :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"T
    HEN740              :rem 36
750 DV=1-7*(A$="D"):IFDV=8THEN
    F$="0:"+F$:OPEN15,8,15,"S"
    +F$:CLOSE15         :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK
    (54)-LEN(T$):POKE782,ZK/25
    6                   :rem 3
762 POKE781,ZK-PEEK(782)*256:P
    OKE780,LEN(T$):SYS65469
                        :rem 109
763 POKE780,1:POKE781,DV:POKE7
    82,1:SYS65466       :rem 69
765 K=S:POKE254,K/256:POKE253,
    K-PEEK(254)*256:POKE780,25
    3                   :rem 17
766 K=E+1:POKE782,K/256:POKE78
    1,K-PEEK(782)*256:SYS65496
                        :rem 235
770 IF(PEEK(783)AND1)OR(191AND
    ST)THEN780          :rem 111
775 PRINT"{DOWN}DONE.{DOWN}":G
    OTO310              :rem 113
780 PRINT"{DOWN}ERROR ON SAVE.
    {2 SPACES}TRY AGAIN.":IFDV
    =1THEN720           :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E
    2$:PRINTE1$;E2$:CLOSE15:GO
    TO720               :rem 103
790 PRINT"{CLR}{RVS}*** LOAD *
    **{2 DOWN}"         :rem 212
795 PRINT"{2 DOWN}{PRESS {RVS}
    RETURN{OFF} ALONE TO CANCE
    L LOAD}"            :rem 82
800 F$="":INPUT"{2 DOWN} FILEN
    AME";F$:IFF$=""THENPRINT:G
    OTO310              :rem 144
810 PRINT:PRINT"{2 DOWN}{RVS}T
    {OFF}APE OR {RVS}D{OFF}ISK
    : (T/D)"            :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"T
    HEN820              :rem 34
830 DV=1-7*(A$="D"):IFDV=8THEN
    F$="0:"+F$          :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK
    (54)-LEN(T$):POKE782,ZK/25
    6                   :rem 2
841 POKE781,ZK-PEEK(782)*256:P
    OKE780,LEN(T$):SYS65469
                        :rem 107
845 POKE780,1:POKE781,DV:POKE7
    82,1:SYS65466       :rem 70
850 POKE780,0:SYS65493 :rem 11
860 IF(PEEK(783)AND1)OR(191AND
    ST)THEN870          :rem 111
865 PRINT"{DOWN}DONE.":GOTO310
                        :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.
    {2 SPACES}TRY AGAIN.{DOWN}
    ":IFDV=1THEN800     :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E
    2$:PRINTE1$;E2$:CLOSE15:GO
    TO800               :rem 102
1000 REM BUZZER         :rem 135
1001 POKE54296,15:POKE54277,45
    :POKE54278,165      :rem 207
1002 POKE54276,33:POKE 54273,6
    :POKE54272,5        :rem 42
1003 FORT=1TO200:NEXT:POKE5427
    6,32:POKE54273,0:POKE5427
    2,0:RETURN          :rem 202
2000 REM BELL SOUND     :rem 78
2001 POKE54296,15:POKE54277,0:
    POKE54278,247       :rem 152
2002 POKE 54276,17:POKE54273,4
    0:POKE54272,0       :rem 86
2003 FORT=1TO100:NEXT:POKE5427
    6,16:RETURN         :rem 57
3000 PRINTC$;"{RVS}NOT ZERO PA
    GE OR ROM":GOTO1000
                        :rem 89
```

©

ATARI

ATARI is a trademark of ATARI INC.

Atari 130 XE ... Call for Latest
Atari 1050 Disk Drive ... Price

SUPERPRINTER PACKAGES

SG-10 & U-Print A	289
Panasonic 1091 and U-Print A	339
Panasonic 1090 and U-Print A	269
Legend 880 and U-Print A	287
Citoh 8510AP and U-Print A	369
Powertype & U-Print A	369

Super Printer Packages have no extra charges when shipped in Continental USA.

ATARI INTERFACES

Aid Interfast I	99.95
U-Print A	59.95
MPP 1150	64.95
U-Print A/16K Buffer	79.95
U-Print A/32K Buffer	99.95
U-Print A/64K Buffer	109.95
R-Verter Adaptor	39.95

INDUS GT

for Atari 219

MPP 1000E

Modem 89.95

NEW
U-Call Modem Adapter 44.95

ATARI SUPER SPECIALS

Basic XL-Cart 49.95
Mac 65XL-Cart 62.95
Action-Cart 62.95
Writer's Tool-Cart 62.95
Monkey Wrench II-Cart 23.95
Ramrod XL & Dmniview XL . 99.95

Super Special Prices Effective Now Thru August 31. 1985

ATARI SOFTWARE

BRODERBUND

Bank Street Writer-D	49.95
Loderunner-D	23.95
Spelunker-D	20.95
Stelth-D	20.95
Print Shop-D	34.95
Serpent's Star-D	27.95
Print Shop Paper	16.95

ELECTRONIC ARTS

Archon	19.95
Archon II	24.95
Mule	19.95
Realm/Impossibility	19.95
Murder/Zinderneuf	19.95
Music Construction	19.95
Pinball Construction	19.95
One on One	24.95
Seven Cities of Gold	24.95
Financial Cookbook	29.95

INFOCOM

Hitchiker-D	29.95
Zork I,II or III	29.95
Enchanter-D	29.95
Deadline-D	34.95
Wisbringer-D	29.95

MICROPRDSE

F-15 Strike Eagle-D	23.95
Crusade in Europe-D	27.95
Decision in Desert-D	27.95
Kennedy Approach-D	23.95

MISCELLANEDUS ATARI

Flight Simulator II-D	34.95
S.A.M.-D	41.95
Megafont-D	19.95
Monkey Wrench II-Cart	29.95
Beachhead-D/T	23.95
Spy vs. Spy-D	23.95
MPP Modern Driver-D	19.95
Net Worth-D	49.95
Omnitrend Universe-D	69.95
Adventure Writer-D	27.95
MMG Compiler-D	65.55
Summer Games-D	27.55
Pitstop II-D	27.55
Ultima I-D	23.95
Ultima II-D	41.95
Ultima IV-D	41.95

MISC. ATARI (cont'd.)

Letter Perfect/Spell-D	49.95
Raid Over Moscow-D	27.95
Micro-Leag. Baseball-D	29.95
Paper Clip-D	39.95
On-Track Racing-D	20.95
Home Pak-D	34.95
B-Graph-D	49.95
Data Perfect-D	49.95

OSS

MAC/65 Tool Kit-D	27.95
Action Tool Kit-D	27.95
Basic XL Tool Kit-D	27.95

SSI

Carrier Force-D	41.95
Combat Leader-D/T	27.95
Cosmic Balance II-D	27.95
Cosmic Balance-D	27.95
Broadsides-D	27.95
War in Russia-D	55.95
50 Mission Crush-D	27.95
Duestron-D	34.95
Rails West-D	27.95
Computer Ambush-D	41.95
Computer Baseball-D	27.95
Breakthrough Ardennes-D	41.95
Field of Fire-D	27.95
Imperial Galactian-D	27.95
Comp. Quarterback-D	27.95
Oper. Market Garden-D	34.95
Kamptgrupee-D	41.95

SYNAPSE

Syntile-D	34.95
Syncalc-D	34.95
Syntrend-D	34.95
Synchron-D	27.95
Syncomm-D	27.95
Synstock-D	27.95
Relax-D	79.95
Blue Max 2001-D	23.95

HUNDREDS OF ITEMS AVAILABLE, PLEASE CALL

T-Cassette
D-Disk
Cart-Cartridge

GENERAL HARDWARE

star MICRONICS
THE POWER BEHIND THE PRINTED WORD.

SG-10 Printer... 219

P.O. the Party Quiz . 29.95
For Apple, C-64 & Atari

PRINTER BUFFERS

Microfazer	Call
MPP Micro Stuffer	119.95
U-Print-16K	79.95
U-Print-32K	99.95
U-Print-64K	109.95

MODEMS

Volksmodem 300	59
Volksmodem 1200	189
Hayes 1200	Call
Hayes 300	169
Micromodem IIE	Call
Maxwell 300/1200	319

PRINTERS

Citoh 8510AP	299
Legend 1080	239
Citoh 7500AP	219
Epson	Call
Toshiba 1351	Call
Toshiba 1340	Call
Legend 880	219
Panasonic 1090	199
Panasonic 1091	269
Powertype	309

ATARI AND C-64 SUPER SPECIALS

BUY ANY INFOCOM TITLE AND GET THE CORRESPONDING INFOCOM INVISICLUE ABSOLUTELY *FREE*.

Super Special Prices Effective Now Thru August 31 1985

APPLE

BRODERBUND

Print Shop-D	33.95
Karateka-D	21.95
Champ Loderunner-D	21.95
Bank Street Writer-D	44.95
Print Shop Graphics-D	17.95
Dazzle Draw	39.95

SSI

Gemstone Warrior	21.95
Kamptgruppe	39.95
Phantasie	25.95
Questron	33.95

APPLE MISCELLANEOUS

Ghostbusters	25.95
Flight Simulator II	34.95
F-15 Strike Eagle	21.95
Summer Games	25.95
Micro League Baseball	23.95
Star League Baseball	23.95
Wizardry/Proving	33.95
Ultima III	39.95
Bruce Lee	25.95
Millionaire	33.95
Death/Carribean	21.95
Felony	21.95

INFOCOM

Hitchiker	29.95
Zork I, II or III	29.95
Enchanter	29.95
Deadline	34.95
Wisbringer-D	29.95

ELECTRONIC ARTS

Skyfox	29.95
One on One	29.95
Seven Cities/Gold	29.95
Archon	24.95
Adv. Construction Kit	37.95
Conan	25.95
Sargon III	33.95
Dollars & Sense	64.95
PFS File	79.95
PFS Write	79.95
PFS Report	79.95
Multiplan	84.95
Supercalc III	124.95
Sideways	39.95
Crosstalk	129.95
Jane w/Mouse	129.95
BPI Software	Call

EST. 1982

ComputAbility

P.O. Box 17882, Milwaukee, Wi 53217
ORDER LINES OPEN
Mon-Fri 11 a.m. - 7 p.m. CST • Sat. 12 p.m. - 5 p.m. CST

To Order Call Toll Free

800-558-0003

For Technical Info, Order Inquiries, or Isr Wisc. Orders

414-351-2007

ORDERING INFORMATION: Please specify system. For fast delivery send cashier's check, money order or direct bank transfers. Personal and company checks allow 2 weeks to clear. Charges for COD are $3.00 extra. Purchase Orders welcome in CONTINENTAL USA, include $3.00 shipping per software order. Include 4% shipping on all Hardware orders, minimum $4.00. Mastercard & Visa please include card # and expiration date. WI residents please add 5% sales tax. HI, AK FPD, APO, Canadian orders — add 5% shipping, minimum $5.00. All other foreign orders, please add 15% of purchase, minimum $10.00. All goods are new and include factory warranty. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call 414-351-2007 to obtain an RA# or your return will NOT be accepted for replacement or repair. Prices and availability are subject to change without notice.

COMMODORE 64

SUPER PRINTER PACKAGES

SG-10 & G-WIZ	282
Panasonic 1091 & G-WIZ	333
Panasonic 1090 & G-WIZ	261
Legend 880 & G-WIZ	271
Legend 1080 & G-WIZ	291
Citoh 8510AP & G-WIZ	364

Super Printer Packages have no extra charges added when shipped in Continental USA

MODEMS

Volksmodem 300	59.95
Total Tele Modem	49.95
Mighty Mo	69.95
1660 Modem	49.95

INDUS

Indus GT for C-64 239

Animation Station

Animation Station	59.95
for C-64	59.95

PRINTER INTERFACES

Axiom	69.95
G-Wiz	59.95
Data Share PPI	49.95
Cardco G +	Call

C-64 SUPER SPECIALS

Slap Shot Hockey-D ... 14.95
Mastering The Sat-D . 49.95
The Perfect Score-D ... 43.95
Mirage Database Manager/ Report Generator-D . 49.95
Mirage Professional Word Processor-D ... 49.95

Super Special Prices Effective Now Thru August 31. 1985

COMMODORE SOFTWARE

ACCESS

Beachhead-D/T	23.95
Raid/Moscow-D/T	27.95
Scrolls of Abadon-D/T	23.95
Mach V-Cart	23.95
Beachead II-D	27.95

BATTERIES INCLUDED

Consultant-D	64.95
Paperclip w/Spellpak-D	79.95
Super Busscard II	129.95
Home Organizer Rems-D	16.95
B.I. 80 Card	109.95
Cal-Kit	34.95
Home Pak-D	34.95

CDMMDDORE

Easy Calc-Cart	34.95
Easy Mail-D	19.95
Easy Script-D	44.95
Easy Spell-D	19.95
Logo-D	49.95
The Manager-D	34.95
Suspended-D	29.95
Starcross-D	29.95
Deadline-D	29.95
Cutthroats-D	29.95
The Suspect-D	29.95
Zork I,II or III-D	29.95

ELECTRDNIC ARTS

Sky Fox-D	24.95
Mail Order Monster-D	24.95
Adv. Construction-D	29.95
Racing Destruction-D	24.95

See Atari Section for Rest of Items and Prices

INFDCDM

Enchanter-D	24.95
Planetfall-D	24.95
Sorcerer-D	29.95
Hitchiker-D	24.95
Wisbringer-D	29.95

SSI

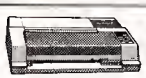
Carrier Force-D	41.95
Questron-D	27.55
Broadsides-D	27.55
Comp. Quarterback-D	27.55
Computer Ambush-D	41.55
Kamptgruppe-D	41.95
Oper. Market Garden-D	34.95
Gemstone Warrior-D	23.55
Imperium Galactium-D	27.55
Phantasie-D	27.95
Cartels/Cutthroats-D	27.95

MISC. COMMODORE

Ultima III-D	41.95
Flight Simulator II-D	34.95
Star Leag. Baseball-D/T	23.95
Mastertype-D/Cart	27.95
Strip Poker-D	23.95
Sam-D	41.95
Spelunker-D	20.95
Stealth-D	20.95
Bungling Bay-D	20.95
Dr. Creep-D	20.95
On-Field Tennis-D/T	23.95
F-15 Strike Eagle-D/T	23.95
Movie Maker-D	37.95
VIP Terminal-D	29.95
Doodle-D	27.95
Ghost Busters-D	24.95
Raids Printer-D	19.95
Karateka-D	20.95
Print Shop-D	34.95
Amazon-D	22.95
Rendezvous/Rama-D	22.95
Farenheit 451-D	22.95
Dragonworld-D	22.95
Shadow Keep-D	22.95
Spy vs. Spy-D	23.95
Stress Relax Sys.-D	79.95
Blue Max-D/T	23.95
Ultima IV-D	41.95
Ultima II-D	41.95
Micro-League Baseball-D	29.95
Home Pak-D	34.95
Math Blaster-D	34.95
Word Attack-D	34.95
Speed Reader II	49.95
Stunt Flyer-D	Call
Sargon III-D	34.95
Super Zaxxon-D	23.95
Decision/Desert-D	27.95
Kennedy Approach-D	23.95
Crusade/Europe-D	27.95
Halley Project-D	29.95
Transylvania-D	23.55
Bounty Bob-Cart	34.55
Blue Max 2001-D	23.55
Fleet System II-D	45.55
Evelyn Wood-D	34.95
Write Now-Cart	34.55
Bruce Lee-D	27.55
Conan-D	27.55
Fast Load-Cart	25.55
Net Worth-D	49.55

No surcharge for MasterCard or Visa

Lyco Computer Marketing & Consultants



SAVE ON THESE IN STOCK PRINTERS

AXION
GP 550 AT (Atari)..... 249
GP 550 CD (C-64)..... 249
GP 550 PC (IBM)..... 239
GP 550 AP (Apple)..... 279
GP 700 AT (Atari)..... 459
GP 700 AP (Apple)..... 459
Elite 5CD (C-64)..... 329

BLUE CHIPS
M12010..... $275
M12010 C-64..... $275

C. ITOH
Prowriter 8510 AP..... 279
8510 BC2..... 389
8510 BP1..... 319
8510 SP..... 379
8510 SR..... 429
8510 SCP..... 459
8510 SCR..... 479
7500 AP..... 205
7500 AP..... 245
1550 P..... 449
1550 BCD..... 489
A-10-20 P..... 459
F 10 40 PU or RDU..... 888
F10 SSPU or RDU..... 1069

CARDCO
LQ1..... 369
LQ3..... 279

CITIZEN
MSP 10..... 329
MSP 15..... 499
MSP 20..... 479
MSP 25..... 599

COMREX
CR II-EC Comriter IIE Parallel..... 359
CR II-ES Comriter II E Parallel..... 379
CR IV-C Comriter IV Parallel..... 689
CR IV-S Comriter IV Serial..... 689

Corona
LP300 Laser Printer 2699
200361 Toner Cartridge ... 89

DIGITAL DEVICES
16K printer buffer 99 75
32K printer buffer 119 75
64K printer buffer 169 95

EPSON
FX 80 + ... 249
LX 80 239
FX 100 579
JX 80 499
LQ 1500 P 1089
LQ 1500 S 1149
Hi-80 color plotter 399

JUKI
Juki 6100 379
RS 232 Serial Board 55
Tractor 119
Sheet Feeder 209
Juki 6300 769

LEGEND
880 219
1080 239
1200 249

★ PRINTER ★
INTERFACING
Available

MANNESMANN TALLY
Spirit 80 255
MTL-160L 549
MTL-180L 739

NEC
NEC 8025 ..... $699
NEC 8027 ..... $359

OKIDATA
Okimate 10 179
182 249
84 645
92 349
93 565
92 Imagewriter 425
92 IBM Version 349

OLIVETTI
DY 250 Parallel 739
DY 250 Serial 729
DY 450 Parallel 1099
DY 450 Serial 1079

PANASONIC
1091 239
1090 189
1092 395
1093 429
3151 459

Smith Corona
Fastext 80 ..... 189 00
D100 ..... 219 00
D200 ..... 399 00
D300 ..... 519 00
L1000 ..... 339 00

STARMICRONICS
SG-10 219
SG-15 379
SD-10 339
SD-15 445
SR-10 489
SR-15 585
Powertype 309
SB-10 ..... CALL

OVER 2000 SOFTWARE TITLES IN STOCK

COMPUTER CARE

BIB
DISK DRIVE CLEANER $12 75
COMPUTER CARE KIT $19 75

NORTRONICS DISK DRIVE CLEANER
with software
REG. 49.95 NOW 19.95

software for IBM-PC Atari, Vic, Apple TL $29 75
DISK CLEANER REFILL $14 75
CASSDRIVE CLEANER ... $9 95
MEDIA BULK ERASER ... $46 75

NEC
PC8201 Portable $429
NECB1 64K Computer System $1049
NECB2 128 K Computer System $1299
PC8221 Thermal Printer ... $ 139
PC8201 8K RAM Chip ... $ 99
PC8206 32K RAM Cart ... $ 299
PC300 Modem ..... $ 65
PC8801 MSDOS 16 Bit Card ..... $ 339

PRINTING PAPER

3000 SHEETS FANFOLD $42 75
1000 SHEETS FANFOLD $19 75
1000 SHEET LETTER ... $21 95
200 SHEETS LETTER ... $8 99
150 RAG STATIONARY ... $10 99
MAILING LABELS (1in) ... $9 95
14 x 11 1000 FANFOLD $24 75

INNOVATIVE CONCEPTS
Flip-n-File 10 ..... 3.50
Flip-n-File 15 ..... 8.25
Flip-n-File 25 Lock 17.95
Flip-n-File 50 ..... 17.25
Flip-n-File 50 Lock 22.95
Flip-n-File Rom 17.25

DRIVES

MSD
SD1 Drive 229
SD2 Drive 469

INDUS
GT Atari 219
GT Commodore 239
GT Apple w/ Controller 219
GT Apple 169

DISKETTES

SKC
(Box 10)
5¼" SKC-SSSD 10.99
5¼" SKC-SSDD 13.99
5¼" SKC-DSDD 15.99

ELEPHANT
(Box 10)
5¼" SSSD 13.99
5¼" SSDD 15.99
5¼" DSDD 19.99

MAXELL
5¼" MD-1 ..... 16.99
5¼" MD-2 ..... 23.75
(Box 10)

IBM-PC SOFT-WARE

*LOTUS
Lotus 1-2-3 ..... 309.00
Symphony 439.00

*ASHTON-TATE
Framework 385.00
d Base II 269.00
d Base III 385.00
d Base II upgrade 135.00

*PARADISE
5 Pak Multifunction 179.00
Modular Graphics 289.00
Modular Graphics Card 315.00

IBM-PC COMPATABLE

CORONA
PPC 22A
Portable 256K Amber 1699
PPC 22G
Portable 256 K-Green 1699
PPCXTA
Portable 256K 10Meg 2899
CORI 28K 128K RAM 1599

Zenith
Z-150 ..... Call

Leading Edge
PC Compatable Call

*LEADING EDGE
Nutshell 69.95
LEWP Basic 65.00
LEWP Merge Print 99.00
LE Spell Correction 169.00

*QUADRAM
Quad Jr Exp. Chassis 519 00
Quad Jr Exp. Memory 209 00
Quad Memory Jr. 209.00
Quadcolor I 209.00

*STB
Super R10 64K 279.00
Super R10 192K 399.00
Super R10 256K 449.00
Graphics Plus II 339.00

*PARADISE
Multi-Display Card 335.00

CONTINENTAL
Home Accountant 85.00

TOLL FREE 1-800-233-8760

MasterCard VISA

TO ORDER

CALL TOLL FREE
800-233-8760
Customer Service 1-717-327-1825 Jersey Shore PA 17740

or send order to
Lyco Computer
P O Box 5088

RISK FREE POLICY

In-stock items shipped within 24 hours of order No deposit on C O D orders Free shipping on prepaid cash orders within the continental U S Volume discounts available PA residents add sales tax APO FPO, and international orders add $5 00 plus 3% for priority mail service Advertised prices show 4% discount for cash, add 4% for MasterCard or Visa. Personal checks require 4 weeks' clearance before shipping. Ask about UPS Blue and Red label shipping. All merchandise carried under manufacturer's warranty. Free catalog with order All items subject to change without notice

ATARI | apple | COMMODORE

ATARI

800 XL Computer ..... CALL
1050 Drive 169
1010 Recorder 44
1020 55
1025 185
1027 239
850 109

SYNAPSE
(ATARI)

Synfile 34.95
Syncalc 34.95
Syncomm 27.95
Syntrend 27.95
Synchron 27.95
Synstock 27.95

SCARBOROUGH

Net Worth 49.95
Mastertype 24.75
Improved Mastertype 24.75
Mastertype's Filer 24.75

SPINNAKER

Delta Drawing Room 19.95
Cosmic Life Room 19.95
Up for Grabs Room 19.95

SSI

Baseball 22.75
Question 26.75
50 Mission Crush 22.75
Broadsides 22.75
Computer Ambush 34.75

Trillium

Shadowkeep $26.75
Fahrenheit 451 ..... $26.75
Amazon $26.75

MICROPOSE

Solo Flight 22.75
NATO 22.75
Spitfire Ace 19.95
F-15 Strike Eagle 22.75

CONTINENTAL

Home Accountant 44.75
1985 Book of Atari Software 16.95

SUBLOGIC

Flight Simulator II 32.75
Night Mission Pinball 18.75

PERSONAL PERIPHEIALS

Super Sketch-Atari 32.95

BRODERBUND

Bank St. Writer 42.75
The Print Shop 29.95
Serpent's Star 24.75
Spelunker 19.95
Stealth 19.95

BUSINESS

VISICALC	$159.75
LETTER PERFECT R	59.00
DATA PERFECT	$89.75
FILE MANAGER	$69.75
HOME FILE MGR	$69.75

ADVENTURE

Dishey 29.95
Ultra Disassembler 29.95

GRAPHIC TABLET

Super Sketch 32.95
Koala Pad 59.95

EASTERN HOUSE

Monkey Wrench II 800 ..... 24.95
Monkey Wrench II 800 XL 24.95

CONTINENTAL

Home Accountant 44.95
1985 Book of Atari Software 16.95

ADVENTURE INTERNATIONAL

Diskey 32.75
Ultra Dissambler 32.75

WICO
(Joysticks)

15-9714 Bat Handle 16.99
50-2002 Super 3-way 19.99
72-4545 Trackball 29.99

Scarborough

Songwriter $24.75
Picturewrit $24.75
Phi Beta F $32.75
Mastertype $24.75
Run I Money $32.75
Net Worth $54.75

Microprose

Solo Flight $22.75
NATO $22.75
Spitfire $22.95
F-15 Strike $22.75
Air Rescue $22.75

SSI

Baseball $22.75
Questron $26.75
Germany 1985 ..... $32.75
50 Missions $21.75

Spinnaker

Alphabet $18.75
Story Machine $19.75
Kids on Keys $18.75
Grandma $19.75
Snooper Troop $22.75

Broderbund

Bank St Writer $42.75
Bank St Filer $42.75
Bank St Mailer $42.75
Bank St Spell $42.75
Mask of Sun $24.95
Print shop 32.95
Lode Runner $22.95

Graphics Tablet

Supersketch $49.95
Kolala $84.95

COMMODORE

Simon's Basic 34.75
Assembler 64 ..... 34.75
Super Expander 22.75
Logo 64 ..... 49.75
Pilot 64 ..... 38.75
Easy Cale 34.75
Easy Script 38.75
C 64 Computer ..... CALL
C 1541 Disk Drive 195
MPS 801 Printer 175
C 1702 Monitor 209
C 1531 Datasette 39.75
C 1660 Auto Modem 85
C 1660 Auto Modem 85

SS1
(C-64)

Computer Baseball 24.75
Field of Fire 24.75
Computer Quarterback 24.75
Questron 24.75
50 Mission Crush 24.75

Scarborough

Songwriter $24.75
Phi Beta F $29.95
Mastertype $24.75
Run I Money $24.75
Net Worth $49.95

SUBLOGIC

Flight Simulator II . 32.75
Night Mission Pinball 22.75

PERSONAL PERIPHERALS

Super Sketch 64 ..... 32.75
Printer Utility 18.75

KOALA
(C-64)

Koala Pad 59.95

CARDCO

C/01 Write Now 29.95
C/02 Write Now · 64 ..... 39.95
D/01 Mail Now · 64 ..... 29.00
D/04 Spell Now · 64 ..... 29.00
D/02 Utility Desk 19.95
CSD-1 Disk Drive(new) ..... CALL
MOD-1 Modern (new) ..... CALL
D/03 Tax Payer (new) 27.95
D/07 Calc Now/64 (new) 27.95
D/08 SUper Printer Utility 27.95
D/08 Super Printer Utility 27.95
CK/1 Numeric Key Pad 34.95
DC/1 Data Cassette 39.95
CB/5 5 Slot Board C-64 ..... 54.00
CR/1 Light Pen 29.75
CE/1 Cassette Interface 29.75
CB/3 3 Slot Board Vic-20 ..... 24.95
CB/6 6 Slot Board Vic-20 ..... 65.00

HES

HES Games 64 ..... 22.95
Omni Writer/Spell 34.95
HES Mon 64 ..... 23.95
Microsoft Multiplan 55.00
Type N Write 19.95
Turtle Graphics II 23.95
Cell Defense 22.95
Paint Brush 12.95
Tri Math 22.95
Graphics Basic 27.95
HES Kit 29.95
Millionaire 23.95
64 Forth 24.95
HES Writer 64 ..... 24.95

HAYES

Smartmodem 300 189
Smartmodem 1200 409
Smartmodem 1200B 379
Micromodem IIE 249
Micromodem 100 269
Chronograph 179
Smart Com II 75

MODEMS

MICROBITS

MPP 1000 E (Atari) 99.00
MPP 1054 (C-64 ..... 69.95

ANCHOR

Volksmodem 55.99
Mark VII 95.99
(auto ans/dial)
Mark VII 259.00
(1200 band)

TELE LEARNING

CM-250 CC-64 ..... 39.95
AP-250 Apple 109.95
IB-250 (IBM) 109.95

NOVATION

J-Cat 89
Smart Cat Plus 319
Smart Cat 103 169
Smart Cat 103/212 389
Auto Cat 209
212 Auto Cat 539
Apple Cat II 239
212 Apple Cat 439
Apple Cat 212 249

CARDCO MOD-1 (C-64) ..... CALL
NESTRIDGE (C-64) ..... CALL
MITEY MO (C-64) ..... CALL
16-0 ALI-E MODEM (C-64)
COMPUSERVE 23.95

BATTERIES INCLUDED

Paper Clip 59.95
Spell Pak 34.95
Consultant 59.95
Paper Clip w/Spell Pak 75.95
Home Pak 34.95
Bus Card 129.95
80 Column Board 109.95

Microprose

Solo Flight $22.75
NATO $22.75
Spitfire $19.95
F-15 Strike $22.75
Air Rescue $22.75

★ *SUPER SPECIAL!* ★

★ *LOWEST PRICES!* ★

MONITORS

AMDEK

300 Green 125
300 Amber 139
310 Amber IBM 155
Color 300 Audio 245
Color 500 Composite 369
Color 600 429
Color 700 495
Color 710 569

PANASONIC

DT 1300 RG1 composite 329

SAKATA

SC-100 Color 219
STS1 Stand 29
SG 1000 Green 99
SA 1000 Amber 109

PRINCETON GRAPHICS

MAX-12 Amber 189
HX-12 RGB 475
SR-12 RGB 599

NEC

JB-1260 Green 95
JB-1201 Green 135
JC 1215 Color 229
JC 1216 RGB 375
JC1460 Color 265
JB-1205 Amber 139

GORILLA

12" Green 78
2" Amber 84

ZENITH

ZVM 122A Amber 84
ZVM 123G Green 75
ZVM 124 Amber IBM 129
ZVM 131 Color 275
ZVM 133 RGB 389
ZVM 135 Composite 449
ZVM 136 Hi Res Color 589

TAXAN

210 Color RBG 249
115 Green 119
116 Amber 125
400 Color RGB 275
410 Color RGB 339
420 Color IBM 429
121 Green IBM 139
122 Amber IBM 145

X-TRON

Comcolor I Composite Green 199

AMERICA'S MAIL ORDER HEADQUARTERS

LYCO COMPUTER

WORLD'S LEADER IN SALES & SERVICE

TO ORDER
CALL TOLL FREE
800-233-8760
In PA 1 717-327-1824
Lyco Computer
P.O. Box 5088
Jeraey Shore, PA 17740

What the world really needs is a 99 cent Double Sided, Double Density Diskette with a LIFETIME WARRANTY!

And DISK WORLD! has it.

Introducing Super Star Diskettes: the high quality diskette with the lowest price and the best LIFETIME WARRANTY!

In the course of selling more than a million diskettes every month, we've learned something: higher prices don't necessarily mean higher quality.

In fact, we've found that a good diskette manufacturer simply manufactures a good diskette...no matter what they charge for it. (By way of example, consider that none of the brands that we carry has a return rate of greater than 1/1,000th of 1 percent!)

In other words, when people buy a more expensive diskette, they aren't necessarily buying higher quality.

The extra money might be going toward flashier advertising, snazzier packaging or simply higher profits.

But the extra money in a higher price isn't buying better quality.

All of the good manufacturers put out a good diskette. Period.

How to cut diskette prices ...without cutting quality.

Now this discovery posed a dilemma: how to cut the price of diskettes without lowering the quality.

There are about 85 companies claiming to be "diskette" manufacturers.

Trouble is, most of them aren't manufacturers.

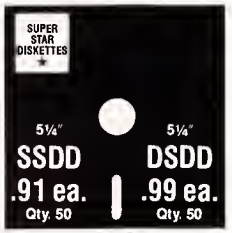
Rather they are fabricators or marketers, taking other company's components, possibly doing one or more steps of the processing themselves and pasting their labels on the finished product.

The new Eastman Kodak diskettes, for example, are one of these. So are IBM 5¼" diskettes. Same for DYSAN, Polaroid and many, many other familiar diskette brand names. Each of these diskettes is manufactured in whole or in part by another company!

So, we decided to act just like the big guys. That's how we would cut diskette prices...without lowering the quality.

We would go out and find smaller companies to manufacture a diskette to our specifications...specifications which are higher than most...and simply create our own "name brand" diskette.

Name brand diskettes that offered high quality at low prices.



Super Star diskettes are sold in multiples of 50 only. Diskettes are shipped with white Tyvec sleeves, reinforced hubs, user ID labels and write-protect tabs.

Boy, did we get lucky. Our Super Star Diskettes are the same ones you've been using for years...without knowing it.

In our search for the low priced, high quality diskette of our dreams, we found something even more interesting.

We found that there are several manufacturers who don't give a hoot about the consumer market for their diskettes. They don't spend millions of dollars in advertising trying to get you, the computer user, to use their diskettes.

Instead, they concentrate their efforts on turning out the highest quality diskettes they can...because they sell them to the software publishers, computer manufacturers and other folks who (in turn) put their name on them...and sell them for much higher prices to you!

After all, when a software publisher or computer manufacturer or diskette marketer puts their name on a diskette, they want it to work time after time, everytime. (Especially software publishers who have the nasty habit of copy-protecting their originals!)

Super Star Diskettes. You already know how good they are. Now you can buy them...cheap.

Well, that's the story.

Super Star diskettes don't roll off the boat from Pago-Pago or emerge from a basement plant just east of Nowhere.

Super Star diskettes have been around for years...and you've used them for years as copy-protected software originals, unprotected originals. Sometimes, depending on which computer you own, the system master may have been on a Super Star diskette. And maybe more than once, you've bought a box or two or more of Super Star diskettes without knowing it. They just had some "big" company's name on them.

Super Star Diskettes are good. So good that a lot of major software publishers, computer manufacturers and other diskette marketers buy them in the tens or hundreds of thousands.

We buy them in the millions.

And than we sell them to you.

Cheap.

When every little bit counts, it's Super Star Diskettes.

You've used them a hundred times...under different names.

Now, you can buy the real McCoy, the same diskette that major software publishers, computer manufacturers and diskette marketers buy...and call their own.

We simply charge less.

Super Special!

Order 50 Super Star Diskettes and we'll be happy to sell you an Amaray Media-Mate 50 for only $8.75, shipping included...a lot less than the suggested retail price of $15.95.

Regular DISK WORLD! price: $10.95 ea. + $2.00 Shpng.

DISKETTE STORAGE CASES

PERFECTOATA DIAL 'N FILE
Terrific! Holds 10 5¼" diskettes. Just flip the lever and they all slide up for easy access and identification. Grey with smoked plastic front.
$2.75 Ea. + .35 Shpng.

DISK CADDIES
The original flip-up holder for 10 5¼" diskettes. Beige or Grey only.
$1.65 ea. + .2D Shpng.

DISKETTE 70 STORAGE
Dust-free storage for 70 5¼" diskettes. Six dividers included. An excellent value.
$11.95 ea. + $3.DD Shpng.

HOURS:
Human: 8AM-6PM Central Time, Monday through Friday
Answering Machine: 6PM-8AM, All Times
MCI MAIL: 24 hours a day.

HOW TO ORDER:

ORDERS ONLY:
1-8DD-621-6827
(In Illinois: 1-312-944-2788)

INQUIRIES:
1-312-944-2788

FOR FASTEST SERVICE, USE ND-CDST MCI MAIL: Our address is DISKWORLD. It's a FREE MCI MAIL letter. No charge to you. (Situation permitting, we'll ship these orders in 24 hours or less.)

SHIPPING: 5¼" & 3½" DISKETTES—Add $3.00 per each 100 or fewer diskettes. **OTHER ITEMS:** Add shipping charges as shown in addition to other shipping charges. **PAYMENT:** VISA, MASTERCARD and Prepaid orders accepted. **COD ORDERS:** Add additional $3.00 special handling charge. **APO, FPO, AK, HI & PR ORDERS:** Include shipping charges as shown and additional 5% of total order amount to cover PAL and insurance. We ship only to United States addresses, except for those listed above. **TAXES:** Illinois residents, add 8% sales tax

MINIMUM ORDER: $35.00 or 20 diskettes.

The Super Star LIFETIME WARRANTY!

Super Star Diskettes are unconditionally warranted against defects in original material and workmanship so long as owned by the original purchaser. Returns are simple: just send the defective diskettes with proof of purchase, postage-paid by you with a short explanation of the problem, and we'll send you the replacements. (Incidentally, coffee stained diskettes and diskettes with staples driven through them don't qualify as "defective".)

WE WILL MEET OR BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES SUBJECT TO THE SAME TERMS AND CONDITIONS.

DISK WORLD!, Inc.

Suite 4806
30 East Huron Street
Chicago, Illinois 60611

Advertisers Index

Technical

Breakthrough

Number 27

Price and performance
no other printer interface can touch!

Only CARDCO would dare improve on its own best seller (the +G has far out-sold any other printer interface, and has set the industry standard by which competitors are judged). The G-Wiz is even better — and costs 30% less.* Our 27th major innovation in Commodore accessories has all the +G's features, and more...

Built-In Buffer for More Speed
The G-Wiz buffer dumps high-resolution screens up to 18 times faster than competitive interfaces without buffers. No more waiting! A complex 50-minute printout now takes just 2:5 minutes with the G-Wiz.

Exclusive! Aspect Ratio Selection
Only the G-Wiz matches graphics characters to standard characters on Okidata, Epson, Star Micronics, Prowriter, Smith Corona, C-ITOH, Gorilla Banana, and many other dot matrix printers. Now you can perfectly align high resolution graphics characters within text blocks, or in columns.

CARDCO excellence triumphs again! The G-Wiz is the "best bang for the buck" on the printer interface market today — and it's backed by CARDCO'S exclusive lifetime warranty! G-Wiz: another distant target for the competition to shoot at.

CARDCO, Inc. 300 S. Topeka/Wichita, KS 67202

* Actual price may vary according to dealer and region.

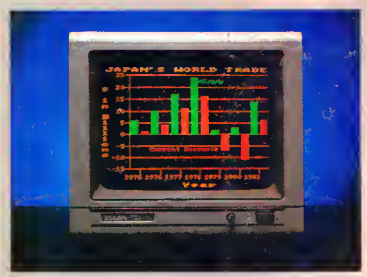
The Wizerds from the Lend of Oz Heve Done It Again!

TAXAN

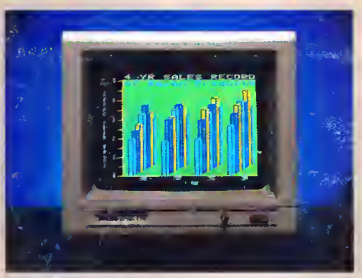
We Make Your Apple Shine

Being #1 is an education. Picking the right Monitor for your Apple takes an educated guess, too!

At TAXAN we keep it simple. Our Apple line of Monitors combines many of the great features of our IBM line. Super High-Res and High-Res capabilities allow for the finest in business graphics. Video games look better, too! And your children will love you for choosing TAXAN instead of Brand X.



MODEL 420 12 inch Super High-Res RGB Color Monitor. Fully compatible with Apple, IBM and most other personal computers. 640x262 line resolution. Unlimited colors available through analog video circuit. Black face, non-glare tube.



MODEL 410 12 inch High-Res RGB Color Monitor. Fully compatible with Apple, IBM and most other personal computers. 510x262 line resolution. Unlimited colors available through analog video circuit.

MODEL 220 14" Composite Color Monitor with Audio
Standard NTSC composite video signal input
Chrominance and luminance inputs available for Commodore, Atari, and other computers
Extremely low geometric distortion
Switchable to green or amber display. Built-in 2 watt audio
Optional: Tilt/Swivel Base available **(Model 110-12)**

MODEL 116 12 inch High-Res Amber Monitor. Long Persistance Phosphor with non-glare screen. Fully compatible with Apple, IBM and most other personal computers. 1000 line resolution. More than 20 MHz bandwidth. 2000 character display. Optional Tilt/Swivel Base available Model 110-12.
Green Phosphor available (Model 115).

410-08 RGB interface card for the Apple II, II+, IIe and Franklin Ace 1000 and 1200 Rev. A only.
410-17 RGB Module for Apple IIc.
410-80B 80 Column and RGB interface card Compatible with Apple IIe. Third color cursor for text editing.
410-8064 Same as 410-80 with additional 64K Ram memory and 6 color double High-Res graphics. Text is color changeable.

MODEL 500 PRINTER BUFFER
Centronics compatible parallel interface with 64K memory expandable to 256K. Four function modes include: Manual, Double, Free and Command. Compatible with most printers.
OPTIONS—
Model 599-01 64K Extra Memory Add-on
Model 599-02 Extra parallel Cable

TAXAN

So, get an education. See TAXAN today!
The Smart Choice.

18005 Cortney Ct. City of Industry, CA 91748 **(818) 810-1291**

*Apple is a registered trademark of Apple Computer, Inc. *IBM is a registered trademark of International Business Machines, Inc.

RetroMags

Our goal is to preserve classic video game magazines so that they are not lost permanently.

People interested in helping out in any capacity, please visit us at www.retromags.com.

No profit is made from these scans, nor do we offer anything available from the publishers themselves.

If you come across anyone selling releases from this site, please do not support them and do let us know.

Thank you!